



**TUGAS AKHIR - TE 141599**

**RANCANG BANGUN MODUL SISTEM OTOMASI  
BANGUNAN BERBASIS KONSEP *INTERNET OF THINGS***

Rephando Amstrabena  
NRP 07111440000201

Dosen Pembimbing  
Dr. Ir. Djoko Purwanto, M.Eng.  
Dr. Mahendra Wardhana, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**TUGAS AKHIR - TE 141599**

**RANCANG BANGUN MODUL SISTEM OTOMASI  
BANGUNAN BERBASIS KONSEP *INTERNET OF THINGS***

Rephando Amstrabena  
NRP 07111440000201

Dosen Pembimbing  
Dr. Ir. Djoko Purwanto, M.Eng.  
Dr. Mahendra Wardhana, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**FINAL PROJECT - TE 141599**

***BUILDING AUTOMATION SYSTEM MODULE  
PROTOTYPE BASED ON INTERNET OF THINGS  
CONCEPT***

Rephando Amstrabena  
NRP 07111440000201

Supervisors  
Dr. Ir. Djoko Purwanto, M.Eng.  
Dr. Mahendra Wardhana, ST., MT.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Electrical Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2018



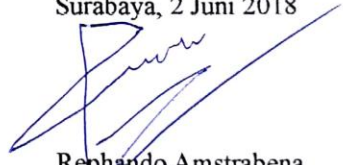
## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “RANCANG BANGUN MODUL SISTEM OTOMASI BANGUNAN BERBASIS KONSEP *INTERNET OF THINGS*” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 2 Juni 2018



Rephando Amstrabena  
NRP. 07111440000201

.....*Halaman ini sengaja dikosongkan*.....



**RANCANG BANGUN MODUL SISTEM OTOMASI  
BANGUNAN BERBASIS KONSEP *INTERNET OF  
THINGS***

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Elektronika  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro**

**Institut Teknologi Sepuluh Nopember**

**Menyetujui :**

**Dosen Pembimbing I**



**Dr. Ir. Djoko Purwanto, M.Eng.**  
**NIP. 196512111990021002**

**Dosen Pembimbing II**



**Dr. Mahendra Wardhana, ST., MT.**  
**NIP. 197204282003121001**



.....*Halaman ini sengaja dikosongkan*.....

# **Rancang Bangun Modul Sistem Otomasi Bangunan Berbasis Konsep *Internet of Things***

Nama : Rephando Amstrabena  
Pembimbing : 1. Dr. Ir. Djoko Purwanto, M.Eng.  
2. Dr. Mahendra Wardhana, ST., MT.

## **ABSTRAK**

Aspek kemudahan dan efisiensi dalam menjalani pekerjaan sehari-hari merupakan sesuatu yang selalu dicari oleh manusia, untuk itu dirancanglah modul sistem otomasi bangunan dalam penelitian ini. Modul ini menggunakan konsep *internet of things*, di mana kontrol utama dari *user* terhadap perangkat keras dari sistem ini terhubung melalui jaringan internet yang diakses melalui aplikasi android.

Sistem otomasi bangunan ini terbagi ke dalam tiga bagian sesuai fungsinya dalam sistem. Bagian pertama adalah *control unit*, bagian ini adalah tempat diaturnya hubungan *input/output* pada sistem yang pola aktivitasnya ditentukan sesuai dengan pengaturan dari *user*. Bagian kedua adalah *command center* yang merupakan tempat untuk *user* memberikan perintah terhadap sistem dalam bentuk aplikasi android. Bagian ketiga adalah *web server* di mana pada sistem ini berfungsi sebagai tempat penyimpanan data kondisi dalam *database* yang menghubungkan antara perintah yang diberikan oleh *command center* dengan *control unit* melalui koneksi internet. Modul ini memiliki dua mode operasi, yaitu mode *auto* dan mode manual. Pada mode operasi *auto*, perangkat yang dikontrol oleh *control unit* pada sistem ini ditentukan oleh keadaan intensitas cahaya dan suhu lingkungannya melalui sensor. Sedangkan pada mode manual, perangkat yang dikontrol ditentukan oleh *user* melalui aplikasi android.

Dari hasil pengujian, disimpulkan bahwa sistem berjalan dengan baik. Akuisisi data oleh sensor suhu pada sistem cukup akurat dengan persen *error* rata-rata sebesar 5,91%, dan sensor cahaya menunjukkan nilai dalam bentuk persentase kemampuan sensor. *Control unit* melakukan *update* data rata-rata setiap 0,88s dalam manual dan setiap 2,87s dalam mode auto. Realisasi dari *control unit* memiliki dimensi 30,7 x 19 x 6,5 cm, dan memakan daya sebesar 5,2W dalam keadaan *standby*.

Kata kunci: *Building Automation, Internet of Things, User-Friendly, Kontrol.*

.....*Halaman ini sengaja dikosongkan*.....

# ***Building Automation System Module Prototype Based on Internet of Things Concept***

*Name* : Rephando Amstrabena  
*Supervisors* : 1. Dr. Ir. Djoko Purwanto, M.Eng.  
2. Dr. Mahendra Wardhana, ST., MT.

## ***ABSTRACT***

*In general people constantly seek for ways to make their daily life easier and cheaper, that is the core reason to design the building automation system module in this research. This module uses the concept of internet of things, where the main control from the user to the hardware of this system is connected through the internet network accessed through an android application.*

*This building automation system is divided into 3 parts according to its function in the system. The first part is the control unit, which is where the input/output relationship on the system is set. The activity pattern is determined by the command of the user. The second part is the command center which is a place where the user can give commands to the system in the form of android application. The third part is the web server which in the system serves as a data storage that connects commands provided by the command center to the control unit via internet connection. This module has two modes of operation, namely auto mode and manual mode which can be selected through the android application. In the auto operating mode, the device controlled by the control unit on this system is determined by the state of the light intensity and the ambient temperature through the sensor. While in manual mode, the controlled device is determined by the user through the android application.*

*From the test results, it is concluded that the system is running well. The data acquisition by temperature sensors on the system is quite accurate with an average error percentage of 5.91%, and the light sensors shows the value in terms of percentage of the sensor capability. On average the control unit updates data every 0.88s in manual mode, and 2.97s in auto mode. The realization of the control unit has dimensions of 30.7 x 19 x 6.5 cm, and takes power up to 5.2W in standby condition.*

***Keywords:*** *Building Automation, Internet of Things, User-Friendly, Control.*

.....*Halaman ini sengaja dikosongkan*.....

## KATA PENGANTAR

Puji syukur atas penyertaan dan berkat yang telah diberikan oleh Tuhan Yang Maha Esa. Karena berkat dan penyertaan-Nya penulis dapat menyelesaikan Tugas Akhir ini. Selama proses penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak baik dukungan secara moril maupun materiil. Oleh karena itu penulis ingin mengucapkan Terima kasih yang sebesar-besarnya kepada berbagai pihak yang mendukung dan membantu dalam tugas akhir ini , diantaranya:

1. Kedua orang tua yang senantiasa memberikan doa, nasihat, motivasi, dukungan dan karena keberadaan merekalah penulis tetap semangat untuk menyelesaikan penelitian ini.
2. Dr. Ir. Djoko Purwanto, M.Eng. dan Dr. Mahendra Wardhana, ST., MT. selaku dosen pembimbing, atas dukungan moriil maupun materiil, serta bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
3. Seluruh dosen-dosen Departemen Teknik Elektro, khususnya dosen-dosen bidang studi Elektronika.
4. Rekan-rekan yang banyak membantu dalam penyelesaian tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Surabaya, 2 Juni 2017

Rephando Amstrabena

.....*Halaman ini sengaja dikosongkan*.....



## DAFTAR ISI

ABSTRAK .....	i
ABSTRACT .....	iii
KATA PENGANTAR.....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xv
BAB I. PENDAHULUAN .....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah .....	3
1.3. Tujuan Penelitian .....	3
1.4. Batasan Masalah .....	3
1.5. Metodologi Penelitian .....	3
1.6. Sistematika Penulisan.....	7
1.7. Relevansi .....	7
BAB II. TINJAUAN PUSTAKA.....	9
2.1. Building Automation System.....	9
2.3.1. Sensor.....	10
2.3.2. <i>Controller</i> .....	10
2.3.3. <i>Output Devices</i> .....	10
2.3.4. Protokol Komunikasi .....	10
2.3.5. <i>User Interface</i> dan <i>Dashboard</i> .....	10
2.2. IoT (Internet of Things).....	10
2.3. Arduino Mega 2560 .....	12
2.3.6. Sumber Daya/ <i>Power</i> .....	12
2.3.7. Memori.....	13
2.3.3. <i>Input/Output</i> .....	13
2.3.4. Komunikasi .....	14
2.3.5. Perlindungan Arus USB .....	14
2.4. Arduino Ethernet Shield.....	15
2.5. Arduino IDE (Integrated Development Environment) .....	16
2.6. DS18B20 (Sensor Suhu) .....	17
2.7. Light Dependent Resistor (LDR) .....	18
2.8. Relay SPDT.....	19
2.9. MIT App Inventor .....	20
2.10. AUTODESK EAGLE 9.0.0 (Student Version) .....	21
2.11. Liquid-Crystal Display (LCD) 20x4 .....	22

BAB III. PERANCANGAN PENELITIAN .....	25
3.1. Perancangan Kuesioner .....	26
3.1.1. Perancangan Pertanyaan Tentang Kebiasaan Masyarakat .....	26
3.1.2. Perancangan Pertanyaan Tentang Ketersediaan Sarana Pendukung Sistem Otomasi Bangunan ....	26
3.1.3. Perancangan Pertanyaan Tentang Kecenderungan Masyarakat Terhadap Teknologi Otomasi Bangunan.....	27
3.2. Diagram Blok Sistem .....	27
3.3. Perancangan Control Unit.....	28
3.2.1. Desain dan Rangkaian Sensor Cahaya.....	28
3.2.2. Desain dan Rangkaian Sensor Suhu .....	30
3.2.3. <i>Sensor Dock</i> .....	32
3.2.4. Desain dan Rangkaian Modul <i>Relay</i> .....	33
3.2.5. Tata Letak dan Koneksi <i>Membrane Keypad 4x4</i> ..	36
3.2.6. Tata Letak dan Koneksi LCD (Dilengkapi Modul I2C).....	37
3.2.7. Arduino Ethernet Shield .....	38
3.2.8. Arduino Mega 2560.....	39
3.2.9. Perancangan Program Utama.....	40
3.2.10. Perancangan Program <i>Internet Connection</i> .....	42
3.2.11. Perancangan Program <i>Settings</i> .....	43
3.2.12. Perancangan Program Operasi <i>Auto-Mode</i> .....	47
3.2.13. Perancangan Program <i>Manual-Mode</i> .....	51
3.4. Perancangan Command Center .....	54
3.3.1. Perancangan <i>Design Interface</i> dan Komponen Aplikasi. ....	54
3.3.2. Perancangan Program Blok Penentuan Mode Operasi.. ....	56
3.3.3. Perancangan Program Blok Kendali Operasi Manual... ..	57
3.5. Perancangan Web Server .....	58
3.3.4. Perancangan <i>Database</i> pada <i>Web Server</i> .....	59
3.3.5. Perancangan Program pada <i>Web Server</i> .....	60
BAB IV. PENGUJIAN, HASIL, DAN ANALISIS .....	63
4.1. Rekapitulasi Hasil Kuesioner .....	65
4.1.1. Rekapitulasi Respon Terhadap Pertanyaan Tentang Kebiasaan Masyarakat.....	65

4.1.2.	Rekapitulasi Respon Terhadap Pertanyaan Tentang Ketersediaan Sarana Pendukung Sistem Otomasi Bangunan .....	66
4.1.3.	Rekapitulasi Respon Terhadap Pertanyaan Tentang Kecenderungan Masyarakat Terhadap Teknologi Otomasi Bangunan .....	67
4.2.	Pengujian dan Kalibrasi Sensor Cahaya .....	68
4.2.1.	Metode Pengujian Sensor Cahaya.....	68
4.2.2.	Proses Konversi Nilai Sensor Cahaya .....	70
4.2.3.	Hasil Pembacaan Sensor Setelah Proses Konversi.....	71
4.3.	Pengujian dan Sensor Suhu .....	71
4.3.1.	Metode Pengujian Sensor Suhu .....	72
4.3.2.	Hasil Pengujian Sensor Suhu.....	74
4.3.3.	Kesimpulan yang Diperoleh Terhadap Penggunaan Sensor.....	75
4.4.	Pengujian Komunikasi Arduino Ethernet Shield dengan Web Server .....	76
4.5.	Pengujian Komunikasi Aplikasi Android dengan Web Server .....	78
4.5.1.	Percobaan Pengubahan Kondisi pada Aplikasi Android.....	78
4.5.2.	Analisis Terhadap Komunikasi Aplikasi Android dengan <i>Web Server</i> .....	81
4.6.	Pengujian Mode Operasi .....	81
4.6.1.	Mode Operasi Manual.....	81
4.6.2.	Mode Operasi <i>Auto</i> .....	83
4.7.	Analisis Keseluruhan Sistem .....	88
BAB V. PENUTUP .....		93
5.1.	Kesimpulan.....	93
5.2.	Saran .....	93
DAFTAR PUSTAKA.....		95
LAMPIRAN A .....		99
LAMPIRAN B .....		105
LAMPIRAN C .....		106
LAMPIRAN D .....		123
BIODATA PENULIS .....		129

.....*Halaman ini sengaja dikosongkan*.....

## DAFTAR GAMBAR

Gambar 2.1	Ilustrasi Building Automation System[6].	9
Gambar 2.2	Aplikasi dari IoT[12].	11
Gambar 2.3	Board Arduino Mega 2560[13].	12
Gambar 2.4	Arduino Ethernet Shield[17].	16
Gambar 2.5	Tampilan Interface dari Software Arduino IDE.	17
Gambar 2.6	Pin Diagram dari DS18B20[19].	17
Gambar 2.7	Diagram Blok IC DS18B20[21].	18
Gambar 2.8	Bentuk Fisik LDR[23].	19
Gambar 2.9	Relay dengan 5 Pin Operasi[25].	19
Gambar 2.10	Skematik Relay 5 Pin Operasi[26].	20
Gambar 2.11	Contoh Pemrograman Berbasis Block Pada MIT App Inventor[28].	21
Gambar 2.12	Logo AUTODESK EAGLE[30].	22
Gambar 2.13	Modul LCD 20x4[32].	23
Gambar 3.1	Diagram Blok Sistem.	27
Gambar 3.2	Sensor Cahaya.	29
Gambar 3.3	Skematik Rangkaian Sensor Cahaya.	29
Gambar 3.4	Diagram Blok Hubungan Sensor Cahaya dengan Arduino pada Sistem.	30
Gambar 3.5	Sensor Suhu DS18B20.	31
Gambar 3.6	Tata Koneksi antara Sensor Suhu dengan Arduino pada Sistem.	31
Gambar 3.7	Sensor Dock untuk Sensor Cahaya dan Sensor Suhu.	32
Gambar 3.8	Skematik Sensor Dock.	32
Gambar 3.9	Modul Relay.	33
Gambar 3.10	Skematik Modul Relay.	35
Gambar 3.11	Peletakkan Membrane Keypad 4x4 dan LCD pada Control Unit.	36
Gambar 3.12	Tata Koneksi antara Membrane Keypad 4x4 dan Arduino Mega 2560.	37
Gambar 3.13	Tata Koneksi LCD dengan Arduino Mega 2560 Menggunakan I2C[26].	38
Gambar 3.14	Tata Letak Ethernet Shield pada Arduino Mega 2560.	39
Gambar 3.15	Desain Peletakkan Arduino Mega 2560 pada Control Unit.	39
Gambar 3.16	Flowchart Program Arduino Utama.	41

Gambar 3.17 Flowchart Program Koneksi Internet pada Arduino. ....	42
Gambar 3.18 Flowchart Program Settings Utama.....	44
Gambar 3.19 Flowchart Program Settings Pengaturan Sensor Cahaya. ....	45
Gambar 3.20 Flowchart Program Settings Pengaturan Sensor Suhu. ....	46
Gambar 3.21 Flowchart Program Operasi Mode Auto. ....	50
Gambar 3.22 Flowchart Program Operasi Mode Manual. ....	53
Gambar 3.23 Desain Interface pada Aplikasi.....	54
Gambar 3.24 (a)Komponen yang Digunakan pada Aplikasi; (b)Gambar yang Digunakan pada Tombol. ....	55
Gambar 3.25 Program Penentuan Mode Operasi. ....	56
Gambar 3.26 Program Kendali Manual. ....	57
Gambar 3.27 (a) gambar file “CH1_OFF.png”; (b) gambar file “CH1_ON.png”. ....	58
Gambar 4.1 Realisasi Desain Fisik dari Control Unit. ....	63
Gambar 4.2 Tata Pengaturan Rangkaian Elektronika pada Control Unit. ....	64
Gambar 4.3 Realisasi Tampilan Aplikasi Android sebagai Command Center.....	65
Gambar 4.4 Pengujian Sensor Cahaya Pada Intensitas Cahaya Maksimal.....	69
Gambar 4.5 Pengujian Sensor Cahaya Pada Intensitas Cahaya Minimal.....	69
Gambar 4.6 Pengujian Sensor Suhu Pada Air tanpa Dipanaskan atau Didinginkan.....	72
Gambar 4.7 Pengujian Sensor Suhu Pada Air yang Dipanaskan. ....	73
Gambar 4.8 Pengujian Sensor Suhu Pada Air yang Didinginkan. ....	73
Gambar 4.9 Grafik Perbandingan Pembacaan Suhu Oleh DS18B20 dengan Thermometer Digital.....	75
Gambar 4.10 Karakter yang Terbaca pada Laman Browser dengan URL: <a href="http://rephanuino.com/BuildingAutomation/viewdbUI.php">http://rephanuino.com/BuildingAutomation /viewdbUI.php</a> .....	76
Gambar 4.11 Data String yang Terbaca oleh Arduino pada URL: <a href="http://rephanuino.com/BuildingAutomation/viewdbUI.php">http://rephanuino.com/BuildingAutomation /viewdbUI.php</a> .....	77
Gambar 4.12 (a) Keadaan Tombol pada Aplikasi; (b) Keadaan Karakter pada Web Server pada Percobaan Pertama.....	79
Gambar 4.13 (a) Keadaan Tombol pada Aplikasi; (b) Keadaan Karakter pada Web Server pada Percobaan kedua. ....	80
Gambar 4.14 Pengujian Operasi Mode Manual Pertama.....	81

Gambar 4.15 Pengujian Operasi Mode Manual Kedua .....	82
Gambar 4.16 Pengujian Operasi Mode Manual Ketiga .....	82
Gambar 4.17 Keadaan Saat Cahaya di Bawah Bottom Limit. ....	84
Gambar 4.18 Keadaan Saat Cahaya di Antara Bottom Limit dan Upper Limit(1). ....	84
Gambar 4.19 Keadaan Saat Cahaya di Atas Upper Limit.....	85
Gambar 4.20 Keadaan Saat Cahaya di Antara Bottom Limit dan Upper Limit(2). ....	85
Gambar 4.21 Keadaan Saat Suhu Ruangan di Atas Upper Limit.....	86
Gambar 4.22 Keadaan Saat Suhu Ruangan di Antara Bottom Limit dan Upper Limit. ....	87
Gambar 4.23 Keadaan Saat Suhu Ruangan di Bawah Bottom Limit. ...	87
Gambar 4.24 Daya Listrik yang Dibutuhkan Oleh Control Unit Sistem. ....	91

.....*Halaman ini sengaja dikosongkan*.....



## DAFTAR TABEL

Tabel 2.1. Spesifikasi Arduino Mega 2560[14]. .....	15
Tabel 2.2. Spesifikasi <i>Relay</i> SPDT 12V.....	20
Tabel 3.1. Spesifikasi <i>Control Unit</i> .....	25
Tabel 3.3. Karakter Representasi Kondisi Relay. ....	51
Tabel 4.1. Respon Pertanyaan Tentang Kebiasaan Responden... ..	66
Tabel 4.2. Respon Pertanyaan Tentang Ketersediaan Sarana Responden.....	67
Tabel 4.3. Respon Pertanyaan Tentang Kecenderungan Responden Terhadap Teknologi Otomasi Bangunan. ....	68
Tabel 4.4. Data Nilai ADC dari Sensor Cahaya. ....	70
Tabel 4.5. Data Nilai Persentase dari Sensor Cahaya. ....	71
Tabel 4.6. Data Pengujian Sensor Suhu.....	74
Tabel 4.7. Data Pengujian <i>Delay</i> Mode Operasi Manual.....	89
Tabel 4.8. Data Pengujian <i>Delay</i> Mode Operasi <i>Auto</i> .....	90

.....*Halaman ini sengaja dikosongkan*.....

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Masalah hemat energi menjadi salah satu topik yang selalu hangat di Indonesia, yang artinya masalah ini belum tertuntaskan. Dalam hal rasio elektrifikasi, Indonesia tergolong negara dengan kategori tertinggal. Rasio elektrifikasi Indonesia pada pertengahan bulan Maret 2018 adalah 95,35%, yang artinya masih sekitar 5% dari seluruh masyarakat Indonesia belum dapat menikmati energi listrik dalam menjalani aktivitas sehari-hari. Dengan rasio elektrifikasi 62,1% di Papua, dan 61,02% di Nusa Tenggara Timur, menunjukkan bahwa ada beberapa daerah di Indonesia yang masih sangat kurang terjamah aliran energi listrik[1]. Artinya metode untuk meningkatkan penghematan energi listrik di Indonesia masih sangat dibutuhkan. Sistem yang memiliki hubungan antara sensor dan aktuator yang dan memiliki hubungan integrasi melalui internet dapat menambah kemungkinan untuk meningkatkan penghematan energi [2].

Sekarang dunia sedang bergerak menuju sistem yang pintar dan akan terus mencari yang lebih pintar. Sistem IoT merupakan jawaban dari pencarian sistem yang lebih pintar tersebut. Dengan sistem IoT, semua perangkat dimaksudkan agar menjadi cukup pintar untuk dapat berkomunikasi dengan perangkat lainnya melalui internet. Komunikasi yang terjalin antar perangkat-perangkat tersebut bertujuan untuk menunjang fungsi otomasi, aksesibilitas, dan fungsi lainnya untuk menyelesaikan masalah lingkungan.

Sistem *internet of things* diprediksikan untuk mencakup *scope* yang sangat luas dengan memiliki 50 miliar obyek yang berkomunikasi melalui internet dalam beberapa jaringan pada tahun 2020[3]. Dengan kata lain penggunaan konsep *internet of things* akan menjadi sesuai yang sangat umum dan mencakup sektor kehidupan manusia secara menyeluruh. Hal ini membuka banyak gerbang terhadap pengembangan teknologi, termasuk juga manajemen pemakaian energi listrik oleh masyarakat.

Dengan hadirnya teknologi *smartphone* saat ini, kemungkinan pada perkembangan konsep *internet of things* menjadi semakin besar dan jelas. *Smartphone* merupakan perangkat yang cukup pintar untuk dapat mengakses dan berkomunikasi lewat jaringan internet, karena

itu sudah sepantasnya konsep *internet of things* diintegrasikan dengan fungsi dari *smartphone* ini. Di Indonesia jumlah pengguna *smartphone* mencapai angka 55 juta pada tahun 2015, dan diprediksi akan mencapai angka 92 juta pada tahun 2019. Angka ini merupakan angka pengguna *smartphone* ketiga terbanyak se-Asia Pasifik[4]. Hal ini menciptakan relevansi terhadap pemanfaatan teknologi *smartphone* dalam pengembangan teknologi di Indonesia.

Dunia pada era ini memiliki masyarakat yang sangat sibuk, dapat dilihat dari aktivitas sehari-hari masyarakat, khususnya di perkotaan. Dalam keadaan ini, masyarakat tentu membutuhkan hal yang dapat membantu melakukan pekerjaan-pekerjaan “sepele” yang kemudian akan memakan tenaga dan waktu.

Kondisi sibuk seperti yang dikatakan sebelumnya juga menambah kemungkinan bagi masyarakat untuk tidak memerhatikan penggunaan energi yang mereka lakukan, biasanya lewat lupaanya mematikan peralatan elektronik seperti AC, televisi, dan semacamnya setelah digunakan. Terlebih lagi untuk melepas hubungan alat elektronik terhadap sumber listrik, hampir semua masyarakat tidak melakukan hal ini dikarenakan anggapan sepele terhadap hal tersebut atau kurangnya informasi terhadap dampaknya. Banyak peralatan elektronik yang tetap menyerap listrik walaupun pada keadaan mati, hal ini dikarenakan beberapa alat elektronik memiliki *standby power*. Keadaan ini biasa disebut dengan istilah “vampir listrik”, karena seringkali muncul tagihan listrik yang berlebih walaupun alat-alat elektronik tidak digunakan[5].

Sistem otomasi bangunan dapat menyelesaikan masalah besar yang dianggap “sepele” tersebut. Dengan sistem yang melakukan tugasnya secara otomatis, pengguna tidak perlu melakukan hal yang membuang waktu dan tenaga untuk menghemat energi listrik, karena semua proses telah berjalan secara digital dan pintar.

## 1.2. Perumusan Masalah

Permasalahan yang dibahas dalam penelitian ini adalah:

1. Perancangan *hardware control unit* untuk sistem otomasi bangunan serta pemrograman *user interface* untuk pengaturan sensor, dan hubungan input/output.
2. Perancangan *software* berupa aplikasi android sebagai *command center* pada sistem.
3. Perancangan *web server* dan sistem komunikasi antara *command center* dengan *control unit*.

## 1.3. Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut:

1. Menghasilkan sistem otomasi bangunan yang *user-friendly* dan dapat digunakan oleh masyarakat umum.
2. Mengaplikasikan konsep *internet of things* pada sistem otomasi bangunan.
3. Mengurangi pemakaian energi listrik secara umum oleh pengguna alat.

## 1.4. Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

1. Sistem hanya dapat digunakan selama *control unit* memiliki akses internet melalui kabel *ethernet*.
2. Kontrol otomatis pada sistem dipengaruhi hanya oleh parameter intensitas cahaya dan suhu lingkungan sekitar.
3. *Control unit* hanya memiliki 8 *channel* keluaran, sehingga pemakaiannya dibatasi oleh angka tersebut.
4. Arus maksimal yang digunakan pada setiap *channel* keluaran terbatas pada kemampuan *relay* yang digunakan yaitu 10A.

## 1.5. Metodologi Penelitian

Langkah-langkah yang dikerjakan pada penelitian ini adalah sebagai berikut:

### 1. Studi literatur

Studi literatur berisi serangkaian kegiatan pengumpulan dan pengkajian dasar teori yang relevan dan terpercaya untuk menunjang perancangan penelitian ini. Literatur dapat bersumber dari paper,

jurnal, artikel, buku, maupun *website*, yang bertaraf nasional dan internasional, serta dari hasil konsultasi dengan dosen pembimbing. Namun, terdapat satu sumber rujukan utama yang telah dilakukan yakni *IEEE Explore*.

## **2. Observasi dan Analisa Masalah**

Pada tahap ini dilakukan pengkajian terhadap jenis-jenis sistem otomasi bangunan yang ada. Analisa masalah-masalah pada sistem dilakukan dengan merujuk kepada kebutuhan dan kecenderungan dari masyarakat. Observasi dan analisa masalah dilakukan dengan mengkaji paper, jurnal, dan isu-isu terkini dari *website* terkait sistem otomasi bangunan. Salah satu poin penting yang didapat adalah pentingnya sistem yang *user friendly* dalam perancangan sistem otomasi bangunan, agar dapat digunakan sesuai keinginan pengguna secara pribadi. Selanjutnya dilakukan inovasi dalam sistem untuk memenuhi aspek tersebut.

## **3. Perancangan Sistem *Building Automation***

Perancangan sistem bertujuan mendapatkan desain dan mekanisme sistem yang optimal dan efisien sesuai dengan fungsi dan kegunaan yang diinginkan. Pada tahap ini dirancang sistem yang terbagi dalam tiga bagian (sub-sistem) yaitu, *control unit*, *command center*, dan *web server*. Ketiga bagian ini memiliki fungsi spesifik masing – masing pada sistem.

*Control unit* adalah bagian yang melakukan fungsi kontrol terhadap keluaran dan masukan pada sistem. *Command center* merupakan bagian sistem yang melakukan fungsi perintah terhadap *control unit*, di mana aktivitas dari *command center* ini langsung diatur oleh *user*. *Web server* merupakan pusat komunikasi yang menerima perintah dari *command center* dan menjadi tempat bagi *control unit* untuk menerima perintah.

## **4. Persiapan Alat dan Bahan**

Pada tahap ini dilakukan persiapan alat-alat dan bahan yang dibutuhkan untuk merealisasikan sistem *building automation* yang telah dirancang. Hal ini dilakukan agar proses pembuatan dan realisasi komponen-komponen pada sistem lebih jelas dan terstruktur, sehingga mengurangi kemungkinan kesalahan rancang yang menyebabkan kerugian. Sistem ini membutuhkan beberapa komponen di dalamnya, beberapa komponen utama yang dibutuhkan antara lain: arduino mega 2560, arduino ethernet shield, LCD 20x4, *membrane keypad* 4x4, relay (dan komponen penunjang rangkaian

*relay module*), light dependent resistor, sensor suhu DS18B20, AC adaptor 12Vdc, kabel ethernet, kabel *power*, body akrilik, dan komponen penunjang lainnya. Pembelian bahan-bahan tersebut dilakukan di: Berkas Elektronik, Isee, Digiware Store, Akhi Shop, dan beberapa toko *online*.

## **5. Realisasi Komponen Sistem *Building Automation***

Pada bagian ini dilakukan proses pembuatan alat dan sistem interkoneksi yang menjadi komponen pada sistem. Proses realisasi ini termasuk perancangan rangkaian elektronika dan desain fisik (perangkat keras), dan perangkat lunak yang dibutuhkan untuk sistem dapat bekerja dengan baik.

Proses realisasi dari *control unit* terdiri dari: pendesainan PCB dengan aplikasi AutoDesk EAGLE untuk *relay module*, sensor cahaya, dan *sensor dock* yang menjadi tempat penghubung sensor dengan *microcontroller* arduino, merancang tata hubungan komponen mulai dari LCD 20x4, *keypad*, arduino, ethernet shield, *sensor dock*, dan *relay module*, desain wadah akrilik yang menjadi tubuh dari *control unit*, hingga yang terakhir adalah perancangan perangkat lunak berupa kode program untuk menentukan pola kerja pada *control unit* yang ditanam pada arduino mega 2560 sebagai *microcontroller* pada sistem.

Setelah itu dilakukan juga realisasi *command center* pada sistem yang bentuknya merupakan aplikasi android yang dapat digunakan melalui *smartphone* dan *gadget* lainnya yang kompatibel. Perancangan dari *command center* dilakukan melalui aplikasi *web based* bernama MIT App Inventor, aplikasi dibuat dengan melakukan desain komponen dan tampilan pada *gadget* dan juga pemrograman dengan basis *block programming* yang melakukan fungsi untuk mengakses dan memberikan perintah pada suatu *website*.

Terakhir dibuat pula *web server* pada sistem, dengan nama *domain website* "<http://rephanuino.com/>". *Web server* pada sistem ini dibuat untuk menerima perintah dari *command center* dan mengubah keadaan suatu indikator sesuai perintah tersebut, indikator tersebut yang nantinya akan dibaca oleh *control unit* dan bekerja berdasarkan keadaan indikator tersebut. Perancangan *web server* ini merupakan perancangan perangkat lunak yang diprogram menggunakan bahasa *Hypertext Preprocessor* (PHP).

Tahap realisasi ini dilakukan pada umumnya di laboratorium elektronika dasar B.202 departemen teknik elektro ITS, dan di tempat

kediaman pribadi penulis.

## **6. Tahap Pengujian**

Pengujian pada sistem ini dilakukan secara per bagian dan keseluruhan. Pertama dilakukan pengujian per bagian sistem, agar memudahkan proses *troubleshooting* bila ada kesalahan yang terjadi. Pada *control unit* diuji hubungan antar komponen terlebih dahulu, setelah hubungan antar komponen sudah baik, maka diuji fungsi – fungsi yang dijalankan oleh *control unit*, antara lain: melakukan fungsi pengaturan lewat *interface* dari *keypad* dan LCD, melakukan fungsi koneksi terhadap *web server* melalui internet dengan kabel ethernet, serta mengolah data dari sensor dan memberikan perintah pada aktuator pada *relay module*. Setelah itu dilakukan pengujian pada *command center*. Pengujian pada *command center* ini meliputi pengujian pengaplikasiannya pada *smartphone* dan pengujian fungsi perintah terhadap *web server*. Dan pada *web server* dilakukan pengujian *database* dan tampilan karakter pada *website*. Setelah itu juga dilakukan pengujian program untuk mengatur kondisi karakter pada *website*.

Pengujian selanjutnya adalah pengujian sistem secara keseluruhan dan fungsi yang dilakukan oleh sistem. Pertama dilakukan pengujian terhadap fungsi operasi manual pada sistem *building automation*, setelah itu dilakukan pula fungsi operasi *auto* pada sistem, sehingga diketahui apakah hubungan dari setiap komponen pada sistem bekerja dengan baik dan seperti keinginan. Pengujian dilakukan di laboratorium elektronika dasar, ruang B.202 departemen teknik elektro ITS.

## **7. Analisa dan Evaluasi**

Analisa dilakukan terhadap hasil pengujian sehingga dapat diketahui *attribute* dari komponen sistem serta karakteristiknya, dari sisi perangkat keras maupun sisi perangkat lunak. Analisa ini dilakukan untuk dapat melakukan perbaikan pada sistem dan komponennya bila ada fungsi yang tidak berjalan sesuai dengan keinginan. Kemudian akan dilakukan evaluasi pada sistem dan komponennya, agar fungsi dari sistem dapat berjalan sesuai dengan keinginan dengan seefektif dan seefisien mungkin.

## **8. Penyusunan Laporan Tugas Akhir**

Tahap penyusunan laporan merupakan tahap terakhir dari proses pengerjaan penelitian ini. Laporan berisi seluruh hal yang berkaitan dengan penelitian yang telah dikerjakan yaitu meliputi pendahuluan,



studi literatur, perancangan dan pembuatan sistem, pengujian dan analisa, dan juga penutup.

## 1.6. Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi Lima Bab dengan sistematika penulisan sebagai berikut:

- Bab I : Pendahuluan  
Bab ini berisi tentang penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.
- Bab II : Tinjauan Pustaka  
Bab ini menjelaskan tentang teori penunjang yang terkait maupun yang dibutuhkan dalam pengerjaan tugas akhir ini. Dasar teori yang menunjang meliputi *Building Automation System*, IoT, Arduino Mega 2560, Arduino Ethernet Shield, Arduino IDE, DS18B20, *Light Dependent Resistor*, *Relay*, MIT App Inventor, AutoDesk EAGLE 9.0.0 (*Student Version*), *Liquid-Crystal Display* (LCD).
- Bab III : Perancangan Penelitian  
Bab ini menjelaskan tentang perencanaan kuesioner dan sistem yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang menjadi komponen sistem *building automation* pada tugas akhir ini.
- Bab IV : Pengujian, Hasil, dan Analisis  
Bab ini menjelaskan tentang hasil kuesioner, pengujian alat dan analisis terhadap hasil yang didapatkan.
- Bab V : Penutup  
Bab ini berisi tentang kesimpulan yang diperoleh terhadap hasil realisasi dari alat dan juga saran untuk pengembangan alat di kemudian hari.

## 1.7. Relevansi

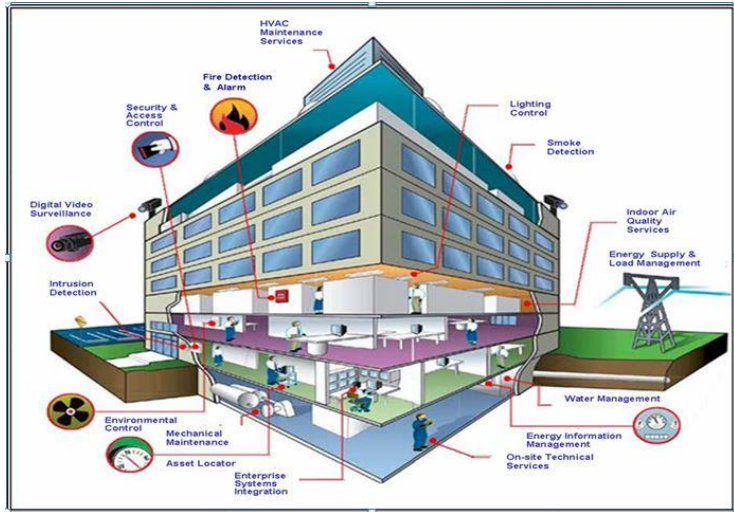
Teknologi sistem *building automation* merupakan teknologi yang sangat relevan dengan kehidupan masyarakat pada era ini, di mana masyarakat semakin sibuk melakukan pekerjaannya dan menumbuhkan kebutuhan akan kemudahan dan kepraktisan dari pengoperasian segala hal. *Smartphone* dan koneksi internet ada di

ujung jari hampir setiap orang terutama di perkotaan, hal tersebut meninggikan tingkat relevansi dari pengaplikasian konsep *internet of things* pada kehidupan masyarakat. Maka dari itu sistem ini menggunakan konsep *internet of things*, yang mana akan membantu kemudahan dalam kehidupan banyak orang khususnya masyarakat di perkotaan.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. *Building Automation System*



**Gambar 2.1** Ilustrasi *Building Automation System*[6].

*Building Automation System* (BAS) adalah sistem dengan kontrol terpusat otomatis untuk fungsi pemanasan, ventilasi dan pendingin ruangan, pencahayaan dan sistem lainnya pada suatu bangunan. Tujuan otomasi bangunan adalah meningkatkan kenyamanan penghuni, efisiensi pengoperasian sistem bangunan, pengurangan konsumsi energi dan biaya operasi, dan menciptakan siklus hidup utilitas yang lebih baik[7]. Otomatisasi bangunan adalah salah satu contoh dari sistem kontrol terdistribusi, yaitu jaringan komputer perangkat elektronik yang dirancang untuk memantau dan mengendalikan sistem mekanis, keamanan, pencahayaan (terutama pencahayaan darurat), sistem kontrol HVAC dan kelembaban dan sistem ventilasi di dalam bangunan[8].

Bangunan yang terintegrasi dengan *building automation system* sering disebut sebagai “bangunan pintar” atau “rumah pintar”[9]. Bangunan industri dan komersil secara historis telah menggunakan

protokol yang terpercaya dalam otomasi bangunan (seperti *BACnet*), dan protokol-protokol swasta seperti “*X-10*” digunakan untuk bangunan-bangunan pribadi.

Fungsi inti dari *building automation system* adalah untuk terus menjaga iklim suatu bangunan dalam kisaran yang ditentukan, memberikan cahaya ke kamar berdasarkan jadwal hunian, memberikan peringatan kerusakan. *Building automation system* harus mengurangi biaya energi dan pemeliharaan bangunan dibandingkan dengan gedung yang tidak dikontrol. Komponen dasar dari BAS meliputi:

#### **2.3.1. Sensor**

Komponen yang berfungsi untuk mengambil informasi dari keadaan sekitar, seperti suhu, intensitas cahaya, kelembaban, dan lainnya untuk menjadi input dari sistem.

#### **2.3.2. Controller**

Merupakan otak dari sistem, komponen ini mengambil *input* data dari sensor pada sistem dan mengatur respon dari sistem berdasarkan data sensor yang didapatkan.

#### **2.3.3. Output Devices**

Komponen ini berperan untuk mengerjakan perintah dari *controller* pada sistem, komponen ini biasanya berupa rele atau aktuator lainnya.

#### **2.3.4. Protokol Komunikasi**

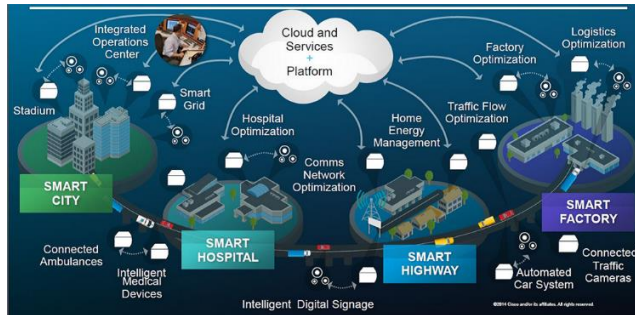
Dapat dikatakan sebagai bahasa yang digunakan di antara komponen-komponen BAS, komponen ini adalah yang menghubungkan komponen-komponen dalam BAS satu dengan yang lain.

#### **2.3.5. User Interface dan Dashboard**

Merupakan komponen yang dibutuhkan sebagai media interaksi antara manusia dengan BAS, juga komponen yang menyajikan data BAS yang telah diambil oleh sensor pada sistem[10].

## **2.2. IoT (Internet of Things)**

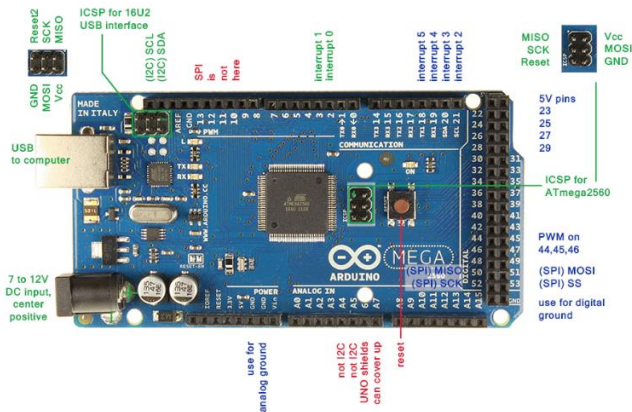
*Internet of Things (IoT)* merupakan konsep sistem jaringan fisik yang terdiri dari komponen elektronik, *software*, sensor, dan terhubung dalam satu jaringan internet, sehingga objek tersebut dapat saling bertukar data[11].



**Gambar 2.2** Aplikasi dari *IoT*[12].

Skema dari interkoneksi antar komponen digambarkan dalam gambar 2.2 di mana aplikasi penggunaan *IoT* dapat digunakan di hampir semua sektor dalam kehidupan manusia. Dengan konsep *IoT* objek-objek tersebut dapat dikontrol oleh suatu alat pengontrol yang terhubung dengan internet, yang kemudian juga dapat terhubung satu dengan yang lain melalui infrastruktur jaringan yang sama, sehingga menciptakan adanya integrasi untuk meningkatkan efisiensi, akurasi, dan faktor ekonomis. Batasan *IoT* sangat luas, salah satunya adalah untuk mempermudah proses bekerja bagi para manusia, seperti halnya mengontrol temperatur dan kelembaban udara suatu ruangan dari belahan dunia lain maupun digunakan untuk pemantauan kesuburan tanah untuk pertanian. Semua hal tersebut dapat dilakukan selama komponen tersebut terhubung pada jaringan internet.

## 2.3. Arduino Mega 2560



**Gambar 2.3** Board Arduino Mega 2560[13].

Arduino Mega 2560 merupakan *board microcontroller* yang menggunakan *processor* Atmega2560. *Board* pada arduino mega memiliki pin *input/output* digital sebanyak 54 buah, dan 14 di antaranya adalah pin *output* PWM. Selain itu juga ada 16 pin analog yang dijadikan sebagai *input* pada *board*. *Board* ini memiliki sebuah *jack power* untuk sumber daya pada *board*, header ICSP, dan sebuah tombol reset. *Board* Arduino Mega 2560 ditunjukkan pada gambar 2.3.

### 2.3.6. Sumber Daya/Power

Arduino Mega 2560 dapat diaktifkan melalui kabel USB atau dengan suplai daya eksternal. *Board* dapat beroperasi pada suplai eksternal dari 6 hingga 20 volt. Rentang tegangan operasi yang disarankan adalah 7 hingga 12 volt karena jika tegangan kurang dari 6 volt maka tidak akan stabil, sedangkan jika lebih dari 12 volt, maka dapat menyebabkan regulator tegangan menjadi panas dan kemungkinan paling buruk adalah dapat merusak *board* Arduino Mega.

### 2.3.7. Memori

Chip Atmega2560 yang terpasang pada Arduino Mega 2560 Revisi 3 memiliki memori 256 KB, dengan 8 KB dari memori tersebut telah digunakan untuk *bootloader*. Jumlah SRAM 8 KB dan EEPROM 4 KB, dimana pada memori ini dapat dilakukan baca-tulis dengan menggunakan EEPROM *library* saat melakukan pemrograman.

### 2.3.3. Input/Output

Pin *input/output* Arduino Mega 2560 jumlahnya yang terbanyak dari semua jenis *board* yang dikembangkan Arduino. Arduino Mega 2560 memiliki 54 pin digital yang dapat digunakan sebagai *input/output*. Pin-pin tersebut bekerja pada tegangan 5 volt. Pada Arduino Mega terdapat beberapa pin yang memiliki fungsi khusus, yaitu:

- Serial: Disini terdapat 4 serial yang masing-masing terdiri dari 2 pin, yakni RX dan TX. (a) Serial 0: pin 0 (RX) dan pin 1 (TX), (b) Serial 1: pin 19 (RX) dan pin 18 (TX), (c) Serial 2: pin 17 (RX) dan pin 16 (TX), (d) Serial 3: pin 15 (RX) dan pin 14 (TX). Pin RX adalah pin untuk *receive* atau menerima, sedangkan TX adalah *transmit* atau untuk mengirim data serial TTL. Pin 0 dan pin 1 adalah pin yang digunakan oleh chip USB-to-TTL Atmega16U2.
- Eksternal *Interrupt*: Pin *interrupt* yaitu pin 2 (*interrupt* 0), pin 3 (*interrupt* 1), pin 18 (*interrupt* 5), pin 19 (*interrupt* 4), pin 20 (*interrupt* 3), dan pin 21 (*interrupt* 2). Pin-pin tersebut dapat dikonfigurasi untuk men-*trigger* sebuah *interrupt* pada nilai “LOW”, tepi naik atau tepi turun, atau merubah nilai. Untuk mengatur *interrupt* tersebut, maka perlu menggunakan fungsi *attachInterrupt()*.
- PWM: Pin 2 hingga 13 dan 44 hingga 46 adalah pin-pin yang menyediakan *output* PWM 8-bit dengan menggunakan fungsi *analogWrite()*.
- SPI: Pin 50 (MISO), 51 (MOSI), 52 (SCK), dan 53 (SS) mendukung komunikasi SPI dengan menggunakan SPI *library*.
- LED: Pin 13, pin ini terhubung dengan *built-in* LED yang dikendalikan oleh pin digital nomor 13. Set HIGH untuk

menyalakan LED, LOW untuk memadamkannya.

- I2C: Pin 20 (SDA) dan Pin 21 (SCL). Pin-pin ini mendukung komunikasi I2C (TWI) menggunakan *wire library*.

Selain pin-pin *digital*, juga terdapat pin-pin *analog* yang berjumlah 16 buah. Masing-masing pin *analog* tersebut memiliki resolusi 10 bits (rentang nilai dari 0 sampai 1024). Secara *default*, pin-pin ini diukur dari *ground* ke 5 volt, namun bisa juga dengan menggunakan fungsi *analogReference()*. Beberapa pin lainnya pada *board* ini yaitu:

- AREF: sebagai referensi tegangan *input analog*.
- Reset: Dengan menghubungkan ke “LOW” untuk melakukan reset terhadap mikrokontroler atau bisa juga dengan menekan tombol reset yang tersedia.

#### **2.3.4. Komunikasi**

Arduino Mega R3 dilengkapi fasilitas untuk berkomunikasi dengan komputer, arduino lainnya, ataupun dengan mikrokontroler lainnya. Chip Atmega 2560 menyediakan komunikasi serial UART TTL (5V) yang tersedia pada pin 0 (RX) dan pin 1 (TX). Chip Atmega16U2 pada *board* berfungsi untuk menerjemahkan bentuk komunikasi melalui USB dan akan tampil sebagai *virtual port* di komputer. *Firmware* 16U2 menggunakan driver USB standar sehingga tidak memerlukan driver tambahan.

Pada Arduino *Software* (IDE) terdapat serial monitor yang memudahkan data tekstual untuk dikirim menuju Arduino atau keluar dari Arduino. Ketika ada data yang ditransmisikan melalui Chip USB to serial via kabel USB ke komputer maka LED TX dan RX akan menyala berkedip-kedip. Pin digital juga dapat digunakan untuk komunikasi serial, yaitu dengan menggunakan *library SoftwareSerial*. Selain itu, Atmega2560 juga mendukung komunikasi I2C (TWI) dan SPI. Pada Arduino *Software* (IDE) telah terdapat *wire library* yang memudahkan pengguna menggunakan bus I2C, serta *SPI library* yang dipakai untuk menggunakan komunikasi SPI.

#### **2.3.5. Perlindungan Arus USB**

Arduino memiliki *polyfuse reset* yang melindungi *port* USB komputer pengguna dari arus pendek atau berlebih. Jika arus lebih



dari 500 mA, maka *fuse* otomatis bekerja.

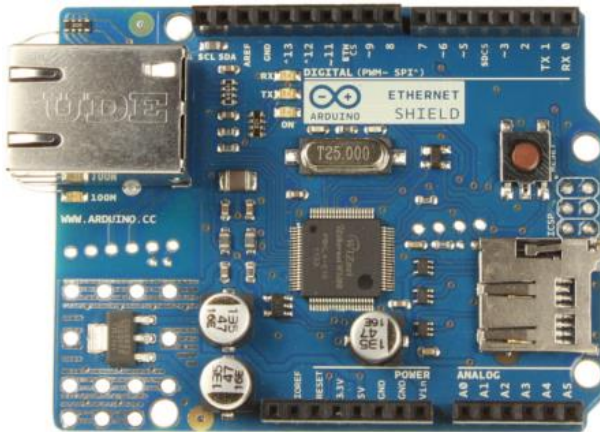
**Tabel 2.1.** Spesifikasi Arduino Mega 2560[14].

No	Parameter	Spesifikasi
1	Mikrokontroler	Atmega2560
2	Tegangan Operasi	5 V
3	Tegangan Input (Rekomendasi)	7-12 V
4	Tegangan Input (Batas)	6-20 V
5	Pin I/O Digital	54 (14 Pin PWM)
6	Pin Input Analog	16
7	Arus DC per pin I/O	40 mA
8	Arus DC untuk pin 3.3 V	50 mA
9	Flash Memory	256 KB (8 KB digunakan <i>bootloader</i> )
10	SRAM	8 KB
11	EEPROM	4 KB
12	Clock Speed	16 MHz

## 2.4. Arduino Ethernet Shield

*Ethernet shield* arduino board berfungsi untuk menambahkan kemampuan arduino untuk dapat terhubung dengan jaringan komputer, bisa ke dalam jaringan lokal ataupun internet. *Ethernet shield* berbasis Wiznet W5100 ethernet chip. *Library* khusus *ethernet shield* digunakan penulisan dan penanaman program ke dalam *board* arduino agar dapat menghubungkan perangkat arduino ke jaringan dengan menggunakan *ethernet shield*. Pada *ethernet shield* terdapat slot *micro-SD*, yang dapat digunakan untuk menyimpan file yang dapat diakses melalui jaringan. Pembaca kartu *micro-SD on-board* diakses menggunakan suatu *library* pemrograman pada arduino IDE. Untuk berkomunikasi dengan W5100 dan kartu SD *arduino board* menggunakan SPI (*Serial Peripheral Interface*). Komunikasi ini diatur oleh *library* pemrograman *spi.h* dan *Ethernet.h*. Bus SPI menggunakan pin digital 11, 12 dan 13 pada Arduino Uno dan pin 50, 51, dan 52 Arduino Mega 2560. Digital Pin 10 digunakan untuk memilih W5100 dan pin digital 4 digunakan untuk memilih kartu SD. Pin yang telah disebutkan sebelumnya tidak dapat digunakan untuk fungsi *general input / output* ketika menggunakan *ethernet shield*.

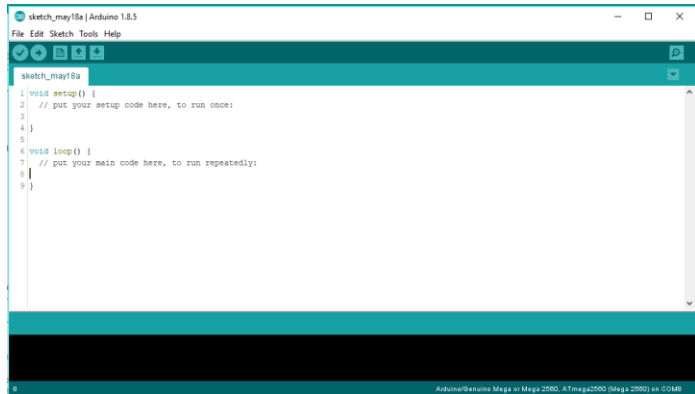
Karena W5100 dan kartu SD berbagi bus SPI, hanya satu yang dapat aktif pada satu waktu[15], [16].



**Gambar 2.4** *Arduino Ethernet Shield*[17].

## **2.5. Arduino IDE (*Integrated Development Environment*)**

Arduino IDE (*Integrated development environment*) pada dasarnya diperuntukkan bagi pengguna arduino untuk melakukan fungsi penulisan program yang tepat dan kompatibel untuk *microcontroller* dengan jenis arduino dan juga menyediakan teknik kompilasi yang mudah. Arduino IDE menyediakan dukungan bahasa untuk *Embedded C* dengan penyediaan *library built-in* untuk membuat penulisan program lebih mudah dan sederhana. Arduino IDE menyediakan *interface* untuk mengunggah program ke dalam *microcontroller* dan juga menampilkan *display* komunikasi serial dari *microcontroller*[18]. *Software* Ini dapat digunakan pada semua perangkat keras arduino, ditambah dengan beberapa perangkat keras *microcontroller* lainnya yang kompatibel.



**Gambar 2.5** Tampilan *Interface* dari *Software* Arduino IDE.

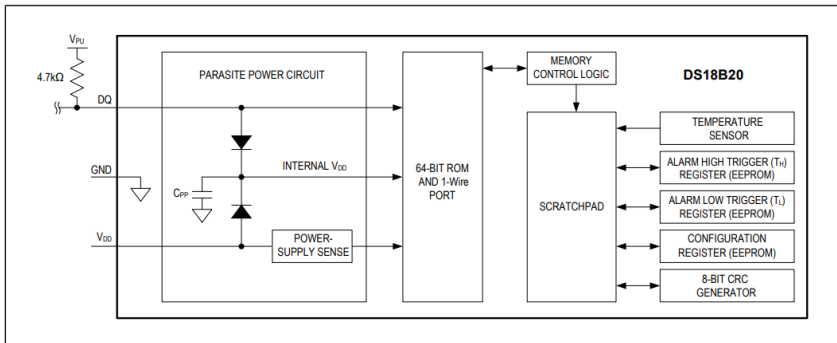
## 2.6. DS18B20 (Sensor Suhu)

DS18B20 merupakan sensor suhu yang dapat bekerja pada udara dan juga air. IC ini memiliki 3 pin hubungan yaitu Vdd, data, dan juga *ground*.



**Gambar 2.6** *Pin Diagram* dari DS18B20[19].

IC ini sudah mencakup bagian-bagian analog, *analog-to-digital converter*, sensor suhu, logika kontrol, ROM (*Read-Only Memory*), dan juga EEPROM (*Electrically Erasable Programmable Read-Only Memory*)[20].



**Gambar 2.7** Diagram Blok IC DS18B20[21].

IC DS18B20 menggunakan protokol komunikasi One-Wire, yang sudah didesain untuk dapat menunjukkan data secara digital berupa nilai suhu dalam satuan celsius atau fahrenheit.

## 2.7. Light Dependent Resistor (LDR)

LDR adalah, sesuai namanya, resistor yang nilai resistansinya berubah-ubah sesuai dengan perubahan intensitas cahaya di sekitarnya. LDR pada dasarnya adalah suatu *photoconductor*, yang mana prinsip kerjanya adalah sebagai berikut: saat cahaya jatuh pada permukaan dari *photoconductor*, terbangunlah pasangan-pasangan *electron-hole* yang menyebabkan peningkatan konduktivitas karena meningkatnya jumlah *carrier*. Untuk *photoconductor* ekstrinsik eksitasi cahaya dapat terjadi di antara tepi *range* dan suatu tingkat energi tertentu pada *gap* energi. Oleh karena itu, akibatnya resistansi dari LDR berkurang seiring dengan bertambahnya cahaya yang ditangkap *photoconductor*[22].



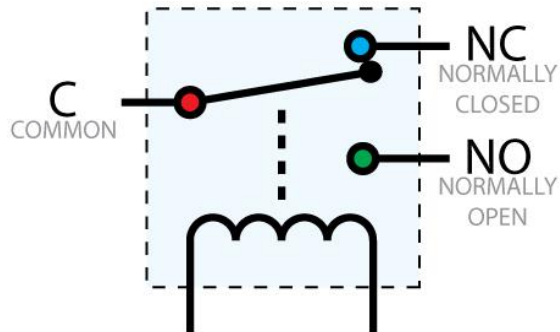
**Gambar 2.8** Bentuk Fisik LDR[23].

## 2.8. Relay SPDT

*Relay* merupakan saklar (*switch*) yang dioperasikan secara elektrik dan merupakan komponen elektromekanikal yang terdiri dari dua bagian utama yaitu elektromagnet (*Coil*) dan mekanikal (seperangkat kontak saklar). *Relay* menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Salah satu fungsi dari *relay* yaitu digunakan untuk mengendalikan sirkuit tegangan tinggi dengan bantuan dari sinyal tegangan rendah atau *relay* dapat mengkonversi keluaran tegangan dari mikrokontroler sebesar 5V[24].



**Gambar 2.9** Relay dengan 5 Pin Operasi[25].



**Gambar 2.10** Skematik Relay 5 Pin Operasi[26].

**Tabel 2.2.** Spesifikasi *Relay* SPDT 12V.

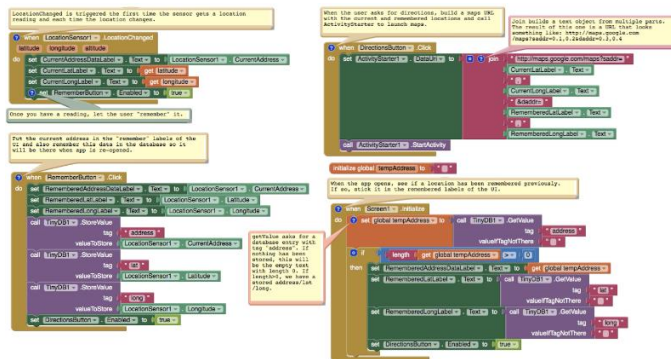
<b><i>Coil Data</i></b>	
<i>Coil Power</i>	360 mW
<i>Nominal Voltage</i>	12 Vdc
<i>Pick-up Voltage</i>	9.0 Vdc
<i>Drop-out Voltage</i>	1.2 Vdc
<i>Maximum Voltage</i>	15.6 Vdc
<i>Coil Resistance</i>	400 ohm
<b><i>Contact Data</i></b>	
<i>Contact Rating</i>	10A 277Vac / 28Vdc
<i>Maximum Switching Voltage</i>	277Vac / 30Vdc
<i>Maximum Switching Current</i>	10A
<i>Maximum Switching Power</i>	2770VA / 210W
<i>Mechanical Endurance</i>	10.000.000 <i>Operations</i>
<i>Electrical Endurance</i>	100.000 <i>Operations</i> ( <i>NO Contact</i> )
<i>Maximum Turn-On Time</i>	10ms
<i>Maximum Turn-Off Time</i>	5ms

## 2.9. MIT App Inventor

Sistem ini memiliki komponen yang dinamakan *command center*, yang mana *command center* ini merupakan pusat perintah utama dari *user* terhadap sistem. *Command center* pada sistem ini berupa aplikasi *smartphone* android yang memiliki fungsi yang

terintegrasi langsung dengan *web server* melalui jaringan internet. Proses pembuatan dari aplikasi *smartphone* ini memanfaatkan *programming environment* yang bernama MIT App Inventor.

MIT App Inventor adalah sebuah *programming environment* visual *web-based* yang intuitif dan *user-friendly* yang dapat digunakan oleh berbagai kalangan untuk menciptakan sebuah aplikasi *smartphone* android[27].



**Gambar 2.11** Contoh Pemrograman Berbasis *Block* Pada MIT App Inventor[28].

MIT App Inventor menggunakan bahasa pemrograman yang berbasis *block* yang bertujuan untuk meningkatkan kemampuan pemrograman dan memperluas jangkauan dari pembelajaran pemrograman, dengan cara menyederhanakan metode dalam melakukan *programming*. Dengan metode pemrograman yang berbasis *block* ini, pengguna dapat melakukan fungsi kompleks dalam aplikasinya dengan waktu yang lebih singkat bila dibandingkan dengan metode bahasa pemrograman berbasis teks yang umum digunakan pada saat ini[29].

## 2.10. AUTODESK EAGLE 9.0.0 (Student Version)

Dalam perancangan dan pembuatan sistem *building automation* pada penelitian ini dibutuhkan pembuatan PCB (*Printed Circuit Board*) untuk rangkaian elektronika dengan komponen-komponen yang dibutuhkan. Pada penelitian ini, desain dari PCB yang

dibutuhkan dilakukan menggunakan *software* EAGLE yang merupakan produk dari AUTODESK.



**Gambar 2.12** Logo AUTODESK EAGLE[30].

EAGLE merupakan *software* yang digunakan oleh banyak kalangan *engineer* dan mahasiswa, karena cara pengoperasiannya yang intuitif dan fitur-fitur yang menunjang perancangan skematik rangkaian dengan mudah dan lengkap. Dengan *software* ini dilakukan perancangan rangkaian untuk modul *relay* yang digunakan pada *control unit* di sistem *building automation* ini.

### **2.11. *Liquid-Crystal Display (LCD) 20x4***

LCD adalah divais untuk menjalankan fungsi *display* yang memanfaatkan sifat dari *liquid crystal* yang dapat memodulasikan cahaya. *Liquid crystal* tidak memancarkan cahaya secara langsung, tetapi menggunakan *backlight* atau *reflector* (pemantul cahaya) untuk menghasilkan gambar dengan warna atau secara *monochrome*[31].





**Gambar 2.13** Modul LCD 20x4[32].

Pada gambar 2.13 diperlihatkan tampilan dari modul LCD berukuran 20x4 yang menampilkan tulisan secara *monochrome*. Modul tersebut dapat dihubungkan secara langsung pada pin digital suatu mikrokontroler atau juga dapat digunakan dengan perantara modul protokol komunikasi.

.....*Halaman ini sengaja dikosongkan*.....

### BAB III PERANCANGAN PENELITIAN

*Prototype* modul sistem *building automation* ini terbagi menjadi tiga komponen besar yang bekerja saling berhubungan untuk menjalankan fungsi utama yaitu mengontrol ruangan pada bangunan secara elektronik. Komponen sistem yang disebutkan sebelumnya antara lain adalah; *command center*, *control unit*, dan *web server*. Dalam pengaplikasiannya, sistem ini dapat menjalankan 2 mode operasi yaitu mode manual dan mode *auto*. Mode *auto* adalah mode di mana fungsi *on/off* pada keluaran ditentukan oleh nilai *input* dari sensor, yaitu sensor cahaya yang memanfaatkan *light dependent resistor* (LDR) dan sensor suhu yang menggunakan LM35, sedangkan mode manual adalah mode di mana fungsi *on/off* pada sistem dikontrol langsung oleh *user* melalui *command center*.

Dalam perancangannya ketiga komponen yang telah disebutkan sebelumnya menggabungkan perangkat keras dan perangkat lunak. *Command center* pada sistem ini merupakan komponen yang memegang kendali utama untuk menentukan tindakan yang dilakukan oleh sistem. *Control unit* berperan sebagai komponen yang mengontrol aktivitas keluaran dari sistem, berdasarkan perintah yang diberikan dari *command center*. Komponen terakhir adalah *web server* yaitu komponen yang berfungsi sebagai tempat penyimpanan nilai kondisi sistem yang diberikan oleh *command center*, dan nilai tersebut yang akan dibaca oleh *control unit* untuk kemudian menjadi dasar kerja sistem, agar sesuai dengan keadaan tertentu pada *web server*.

Spesifikasi *control unit* pada sistem *building automation* yang dirancang pada penelitian ini disajikan pada tabel 3.1 berikut:

**Tabel 3.1.** Spesifikasi *Control Unit*.

Dimensi	30,7 x 19 x 6,5 cm
Jumlah Channel Keluaran	8 Buah
Tegangan <i>Input</i> Maksimal	277VAC, 30VDC
Arus Maksimal Per-Channel	10A
Fitur Operasi	<i>Settings</i> , Mode Manual, Mode <i>Auto</i>
Komponen <i>User Interface</i>	<i>Liquid-Crystal Display</i> 20x4, <i>Membrane Keypad</i> 4x4

Pada bab perancangan sistem ini dijelaskan sistem secara keseluruhan mulai dari perangkat keras hingga perangkat lunak yang dibagi menjadi 3 bagian sesuai dengan komponen sistem yang telah disebutkan sebelumnya, *command center*, *control unit*, dan *web server*.

### **3.1. Perancangan Kuesioner**

Salah satu metode pengambilan data yang digunakan dalam penelitian ini adalah dengan menggunakan kuesioner. Kuesioner dibuat dengan tujuan mendapatkan data mengenai kebiasaan pada masyarakat saat ini, ketersediaan masyarakat akan sarana pendukung sistem otomasi bangunan, dan kecenderungan masyarakat terhadap sistem otomasi bangunan. Sasaran kuesioner yang dibuat adalah kepada masyarakat dengan umur di atas 18 tahun, karena dianggap sudah dewasa dan mengerti akan manajemen energi listrik dan pengeluaran ekonomi.

#### **3.1.1. Perancangan Pertanyaan Tentang Kebiasaan Masyarakat**

Untuk mendapatkan data ini, dibuat bagian yang terdiri dari 4 pertanyaan. Pertanyaan yang dirancang adalah sebagai berikut:

1. “Apakah dalam keseharian anda secara rutin menggunakan peralatan elektronik?”
2. “Peralatan elektronik apa yang sering anda gunakan/nyalakan di rumah atau tempat anda beraktivitas?”
3. “Apakah anda sering lupa mematikan peralatan elektronik yang sudah tidak dipakai di rumah atau tempat anda beraktivitas?”
4. “Apakah alat elektronik yang sedang tidak anda gunakan biasanya tetap terhubung dengan sumber listrik? (termasuk peralatan elektronik yang sudah dalam keadaan mati)”

#### **3.1.2. Perancangan Pertanyaan Tentang Ketersediaan Sarana Pendukung Sistem Otomasi Bangunan**

Dirancang 3 pertanyaan yang mewakili data ini. Pertanyaan tersebut adalah sebagai berikut:

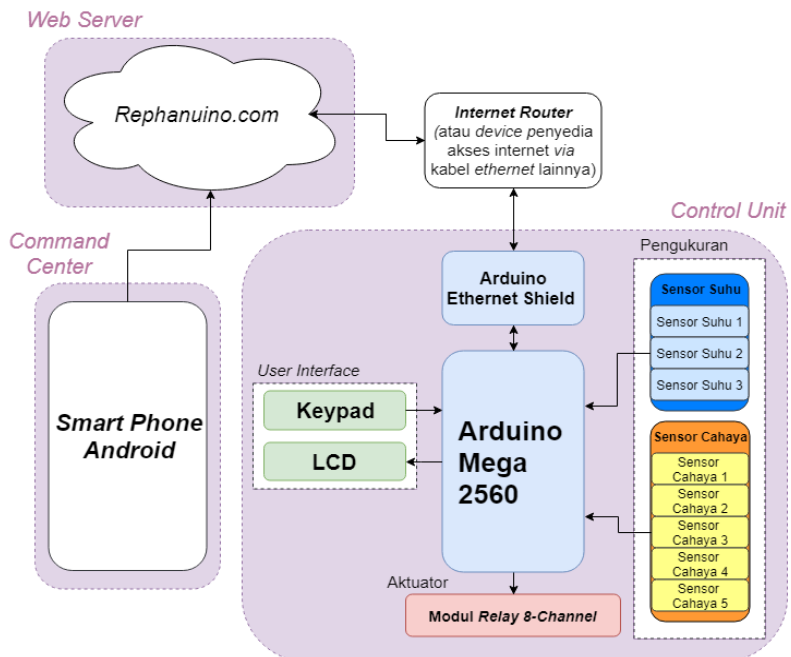
1. “Apakah anda menggunakan smartphone dalam menjalani aktivitas sehari-hari?”
2. “Apakah anda memiliki akses internet di sebagian besar waktu keseharian anda?”
3. “Apakah anda merasa nyaman dalam mengoperasikan smartphone dan internet?”

### 3.1.3. Perancangan Pertanyaan Tentang Kecenderungan Masyarakat Terhadap Teknologi Otomasi Bangunan

Bagian ini terdiri dari 4 pertanyaan yang mewakilinya. Pertanyaan tersebut adalah sebagai berikut:

1. “Apakah anda ingin mengurangi beban biaya listrik anda?”
2. “Apakah anda menginginkan peralatan elektronik di rumah atau tempat anda beraktivitas dapat bekerja secara otomatis sesuai keinginan anda?”
3. “Apakah anda ingin dapat mengendalikan peralatan elektronik di rumah atau tempat anda beraktivitas melalui internet dengan perantaraan aplikasi smartphone?”
4. “Apakah harapan anda terhadap perkembangan teknologi otomasi bangunan?”

### 3.2. Diagram Blok Sistem



**Gambar 3.1.** Diagram Blok Sistem.

Bagian-bagian pada blok diagram sistem pada gambar 3.1 terdiri dari:

(1) *Control Unit*

Bagian yang memiliki fungsi kontrol hubungan *input/output* pada sistem, dan melakukan koneksi terhadap *web server* melalui *website*. Pada bagian ini ada beberapa komponen di dalamnya, antara lain: komponen pengukuran (sensor suhu dan sensor cahaya), komponen aktuator (modul *relay 8-channel*), *user interface* (*membrane keypad 4x4* dan *LCD 20x4*), dan *microcontroller* berupa *arduino mega 2560* dilengkapi dengan *ethernet shield*.

(2) *Command Center*

Bagian ini di dalam sistem berfungsi sebagai pusat bagi *user* memberikan perintah terhadap sistem (khususnya *control unit*). *Command center* pada sistem ini berupa sebuah aplikasi android.

(3) *Web Server*

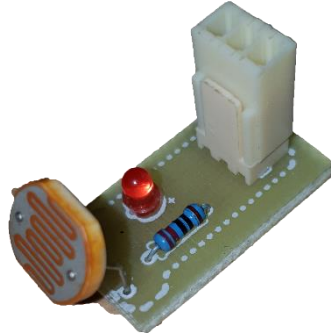
*Web server* pada sistem ini yang berfungsi sebagai penyimpanan data komunikasi dan menghubungkan *command center* dengan *control unit* melalui internet, *web server* pada sistem ini memiliki domain *website* dengan nama “*rephanuino.com*”.

### **3.3. Perancangan Control Unit**

*Control unit* pada sistem ini terdiri dari Sensor suhu dan sensor cahaya sebagai penyedia data *input*, modul *relay 8-channel* sebagai aktuator sistem, *LCD* dan *membrane keypad* sebagai *user interface* untuk fungsi setting sensor, dan *arduino mega 2560* dilengkapi dengan *ethernet shield* sebagai *controller* pada sistem yang terhubung dengan internet melalui hubungan *ethernet*.

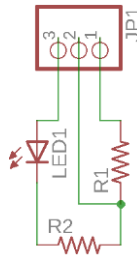
#### **3.2.1. Desain dan Rangkaian Sensor Cahaya**

*Control unit* pada sistem otomasi bangunan ini memiliki 5 *channel* yang disediakan untuk data input dari sensor cahaya. Sensor cahaya yang dirancang pada sistem ini memanfaatkan komponen elektronik *light dependent resistor* (LDR), dimana nilai resistansi dari LDR akan menurun berbanding lurus dengan kenaikan dari intensitas cahaya yang diterima olehnya, yang artinya semakin besar intensitas cahaya yang diterima, maka nilai resistansi dari LDR akan semakin menurun.



**Gambar 3.2.** Sensor Cahaya.

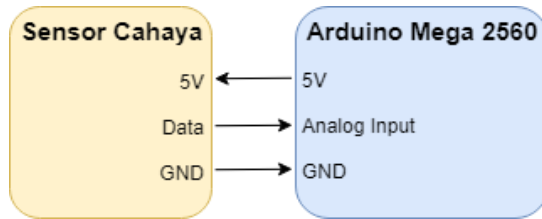
Sensor cahaya pada sistem ini menggunakan tegangan input 5V yang diambil dari pin arduino, resistor 10K $\Omega$ , dan LED merah 3mm sebagai indikator keadaan pada LDR. Bila cahaya LED redup maka cahaya yang diterima oleh LDR juga sedikit, dan bila cahaya LED terang maka cahaya yang diterima oleh LDR intensitasnya tinggi pula.



**Gambar 3.3.** Skematik Rangkaian Sensor Cahaya.

Pada gambar 3.3 R2 adalah LDR dan R1 adalah resistor 10K $\Omega$ . Terlihat bahwa LDR dirangkai secara seri dengan resistor 10K $\Omega$  untuk membuat suatu rangkaian *voltage divider*, nilai tegangan yang diambil pada resistor 10K $\Omega$  dijadikan nilai yang merepresentasikan intensitas cahaya yang dibaca oleh arduino, mengikuti persamaan berikut:

$$V_{out} = \frac{R1}{R1 + R2} \times V_{in}$$



**Gambar 3.4.** Diagram Blok Hubungan Sensor Cahaya dengan Arduino pada Sistem.

Pada gambar 3.4 terlihat bahwa ada 3 hubungan antara sensor cahaya dan arduino. Sensor cahaya mengambil supply tegangan 5V dari arduino, lalu *ground* pada sensor dihubungkan dengan ground pada arduino agar memiliki *common ground*. Pin data pada sensor cahaya merupakan nilai tegangan yang didapat hasil rangkaian *voltage divider* antara LDR dan resistor 10K $\Omega$ . Data dari sensor lalu dihubungkan pada pin analog input di arduino mega 2560 agar dapat dibaca dan diolah.

### 3.2.2. Desain dan Rangkaian Sensor Suhu

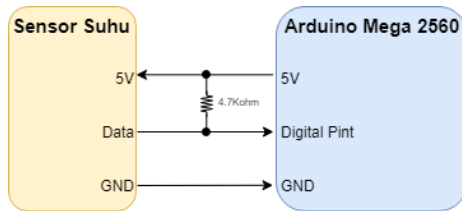
Untuk sensor suhu, *control unit* pada sistem ini menyediakan 3 *channel* yang bisa digunakan untuk membaca nilai suhu. Sensor suhu yang digunakan pada sistem ini adalah DS18B20, sebuah IC yang menghasilkan *output digital* berupa nilai yang berubah-ubah berdasarkan perubahan suhu di sekitarnya, nilai yang dihasilkan merupakan nilai suhu dalam satuan celcius atau fahrenheit. Nilai digital ini bisa didapatkan melalui protokol komunikasi *one wire*. Karena sifat tersebut, maka DS18B20 digunakan sebagai sensor suhu pada sistem ini.





**Gambar 3.5.** Sensor Suhu DS18B20.

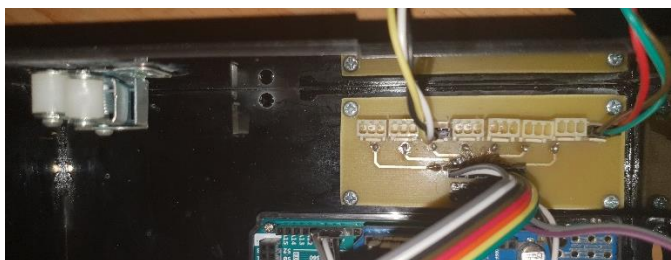
Sensor suhu yang menggunakan DS18B20 pada sistem ini memiliki menggunakan tegangan *supply* sebesar 5V yang diambil dari pin arduino. Keluaran yang dihasilkan dari DS18B20 akan menjadi input digital pada sistem untuk mengetahui keadaan suhu di sekitar sensor. Pada perangkaian sensor dengan arduino ini, dibutuhkan resistor 4.7K $\Omega$  yang menjadi *pull-up* resistor. Resistor dirangkai secara paralel antara jalur supply 5V dan jalur data digital sensor.



**Gambar 3.6.** Tata Koneksi antara Sensor Suhu dengan Arduino pada Sistem.

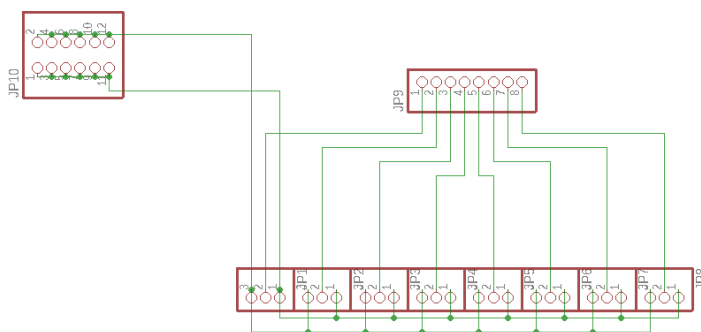
Dari gambar 3.6 terlihat bahwa sensor suhu dengan arduino memiliki tiga macam hubungan pada sistem. Sensor suhu disuplai tegangan oleh arduino melalui pin supply 5V, lalu ground dari sensor suhu dihubungkan ke ground pada arduino agar menjadi satu rangkaian. Output dari sensor suhu, berkomunikasi dengan pin digital, menjadi *input* pada arduino yang kemudian akan dioleh dengan menggunakan protokol komunikasi *one wire*.

### 3.2.3. *Sensor Dock*



**Gambar 3.7.** *Sensor Dock* untuk Sensor Cahaya dan Sensor Suhu.

*Channel* yang disediakan untuk sensor cahaya dan sensor suhu yang ada pada sistem ini berjumlah total 8 (3 untuk sensor suhu, dan 5 untuk sensor cahaya). Setiap sensor membutuhkan suplai tegangan 5V dan *ground* dari arduino mega 2560, sedangkan arduino hanya memiliki pin *supply* 5V, karena itu dibutuhkan rangkaian eksternal untuk menghubungkan pin arduino dengan 8 *channel* sensor tersebut, agar kebutuhan untuk sensor dapat beroperasi terpenuhi.



**Gambar 3.8.** Skematik *Sensor Dock*.

Pada gambar 3.8 diperlihatkan skematik dari rangkaian *sensor dock* yang menghubungkan arduino dengan sensor-sensor pada sistem ini. Rangkaian ini mengambil tegangan 5V dari *board* arduino dan menghubungkannya secara paralel ke 8 *channel* yang akan

digunakan oleh sensor, begitu juga dengan *ground*. Lalu pin data pada sensor akan terhubung pada *pin* di *dock* ini yang sudah terhubung dengan pin analog input pada arduino.

### 3.2.4. Desain dan Rangkaian Modul Relay

Pada *control unit* di sistem ini, digunakan relay sebagai aktuator sistem. Pada modul yang telah dirancang tersedia 8 *channel relay* yang dapat diaktifkan dengan adanya sinyal tegangan 5V yang diberikan dari arduino ke modul *relay*. Setiap *relay* yang digunakan pada modul ini akan aktif dengan diberikan tegangan aktivasi sebesar 12V, sementara sinyal digital yang diberikan oleh arduino hanya sebesar 5V. Untuk itu, pada rangkaian modul *relay*, digunakan IC ULN2803, yang mana fungsinya adalah sebagai driver tegangan. Rangkaian ini di-*supply* tegangan sebesar 12V lewat pin *supply* arduino, memiliki pin untuk *ground*, dan pin sinyal digital 5V sebanyak 8 buah, sesuai dengan jumlah *channel relay* yang dikontrol pada modul ini. Kemudian ada pula pin input AC yang bertegangan 120 hingga 220VAC yang akan menjadi tegangan yang di-*drive* oleh *relay* menuju ke jalur keluaran dari sistem. Bentuk dari *relay module* yang diletakkan pada *control unit* sistem ini dapat dilihat pada gambar 3.9.

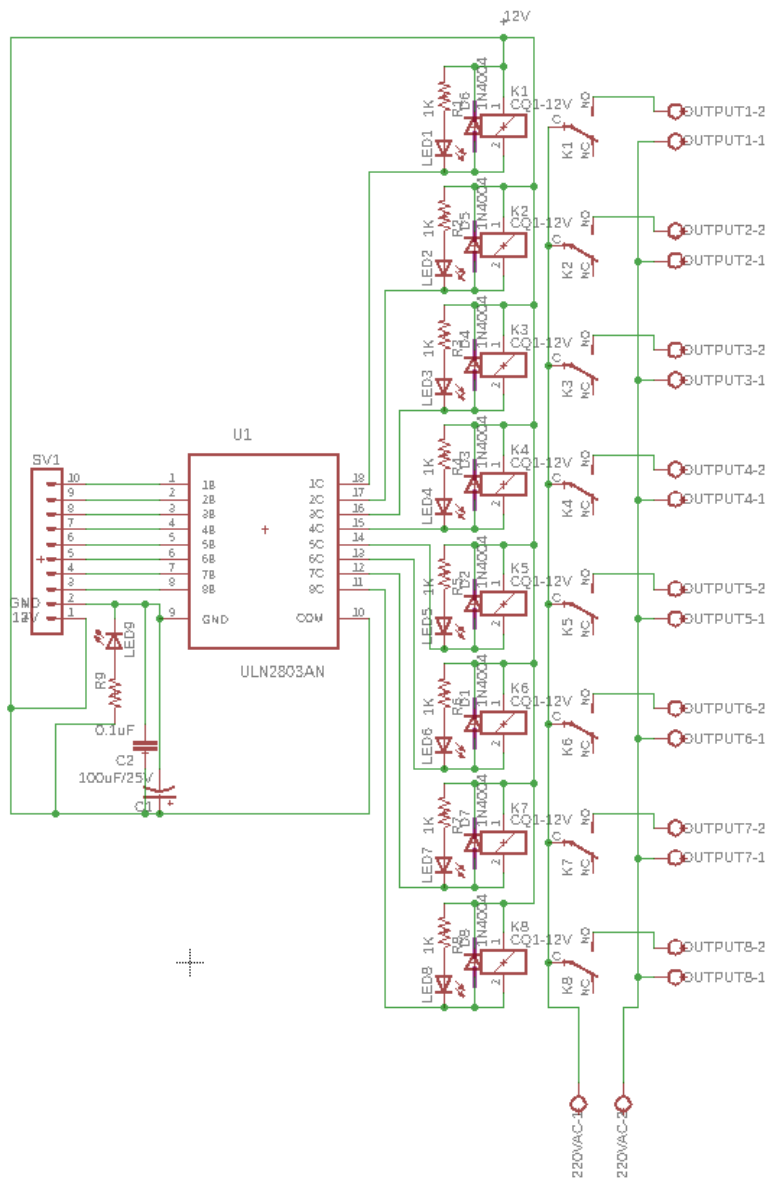


**Gambar 3.9.** Modul Relay.

Pada modul *relay* digunakan 9 buah LED 3mm yang masing-

masing dihubungkan secara seri dengan resistor  $1K\Omega$ . 1 di antara 9 LED tersebut adalah LED yang menjadi indikator aktif atau tidaknya rangkaian *relay module*, lalu 8 LED yang lainnya menjadi indikator aktif atau tidaknya *relay* pada *channel* yang bersangkutan dengannya. Pada rangkaian ini juga digunakan 2 macam kapasitor yang dipasang secara paralel dengan *line supply* 12V dengan *ground* 2 kapasitor ini berfungsi sebagai pengaman pada rangkaian *relay module*.

*Output* dari rangkaian modul *relay* ini berupa tegangan AC 120-220VAC yang di-*drive* melalui *relay* dengan tegangan aktivasi sebesar 12VDC. 2 Line tegangan AC dihubungkan pada *screw terminal* dengan cara membuka *screw* pengunci kabel, memasukkan kabel yang ingin di-*supply* oleh keluaran *relay*, lalu mengunci kembali *screw* yang dibuka agar kabel tidak lepas dari terminal.



**Gambar 3.10.** Skematik Modul Relay.

Pada gambar 3.10 ditunjukkan skematik dari rangkaian modul *relay* pada *control unit* di sistem ini yang didesain menggunakan aplikasi desain PCB yang bernama AutoDesk EAGLE.

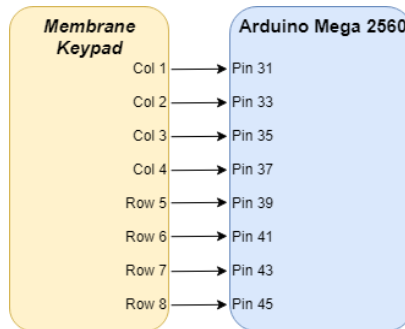
### 3.2.5. Tata Letak dan Koneksi *Membrane Keypad 4x4*

Seperti yang sudah dijelaskan sebelumnya, sistem *building automation* ini menggunakan *keypad* sebagai salah satu alat *user interface* untuk fungsi pengaturan sensor oleh pengguna. *Keypad* yang digunakan adalah *membrane keypad 4x4*, yang adalah *keypad* pipih dengan pin *output* yang berjumlah 8, 4 pin merepresentasikan kolom pada posisi *keypad*, dan 4 pin lainnya merepresentasikan baris pada *keypad*.



**Gambar 3.11.** Peletakkan *Membrane Keypad 4x4* dan LCD pada *Control Unit*.

*Keypad* diletakkan di luar *case control unit* (di permukaan) sehingga dapat dioperasikan oleh *user* secara mudah dan *intuitif*. *Output* dari *keypad* pada sistem ini menjadi *input* pada pin digital di *arduino*. *Keypad* pada *control unit* ini digunakan sebagai *interface user* untuk melakukan pengaturan lewat fitur fungsi “*settings*” yang telah terprogram pada *arduino* sebagai *microcontroller*. Pengaturan yang dilakukan adalah untuk menentukan parameter-parameter yang dibutuhkan pada saat sistem menjalankan mode *auto*. *Input* dari *keypad* ini yang menentukan kerja sistem pada fungsi “*settings*”.

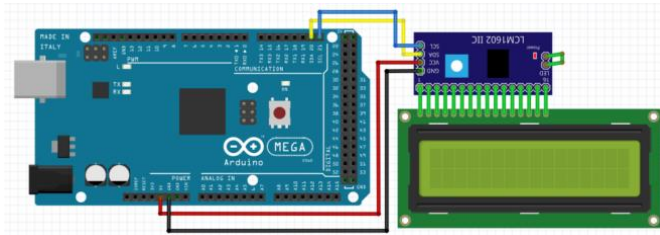


**Gambar 3.12.** Tata Koneksi antara *Membrane Keypad* 4x4 dan *Arduino Mega 2560*.

Seperti yang telah dijelaskan pada bab sebelumnya bahwa setiap tombol yang ditekan pada *membrane keypad* menghubungkan satu pin *column* dengan pin *row* tertentu. Pada sistem ini, hubungan antara pin *column* dan *row* dibaca oleh pin digital 31, 33, 35, 37, 39, 41, 43, dan 45 pada arduino seperti pada gambar 3.12, sehingga hubungannya dapat terbaca dan tombol yang ditekan dapat ditentukan.

### 3.2.6. Tata Letak dan Koneksi LCD (Dilengkapi Modul I<sup>2</sup>C)

Dari gambar 3.11 dapat dilihat bahwa LCD ditempatkan pada permukaan *case* sama seperti *keypad*. LCD pada *control unit* sistem ini digunakan sebagai *user iterface* yang memiliki fungsi untuk menampilkan opsi pengaturan kepada *user* dan juga menampilkan nilai pembacaan dari sensor yang akan diatur, selain itu LCD juga digunakan untuk menampilkan mode yang sedang bekerja pada sistem (*manual* atau *auto*).



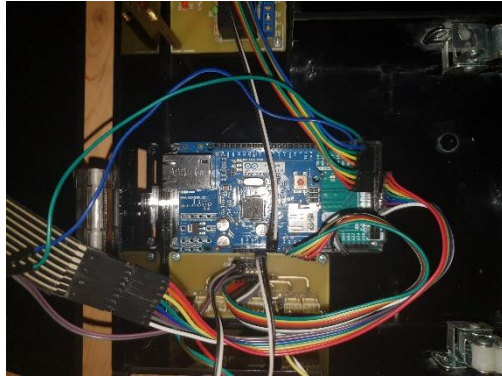
**Gambar 3.13.** Tata Koneksi LCD dengan Arduino Mega 2560 Menggunakan  $I^2C$ [26].

Pada gambar 3.13 terlihat bahwa semua pin pada LCD dihubungkan kepada  $I^2C$  serial adapter yang keluarannya berupa pin SCL (*Serial Clock*) dan SDA (*Serial Data*), juga supply 5V dan ground. Pin SCL dan SDA dari  $I^2C$  serial adapter kemudian dihubungkan dengan pin SCL dan SDA pada arduino sebagai master. Suplai tegangan 5V untuk  $I^2C$  serial adapter diambil dari pin 5V dari arduino dan ground dari  $I^2C$  serial adapter juga dihubungkan dengan grounds arduino.  $I^2C$  serial adapter digunakan pada LCD di sistem ini dengan tujuan mengurangi pin digital arduino yang digunakan oleh sistem ini.

### 3.2.7. Arduino Ethernet Shield

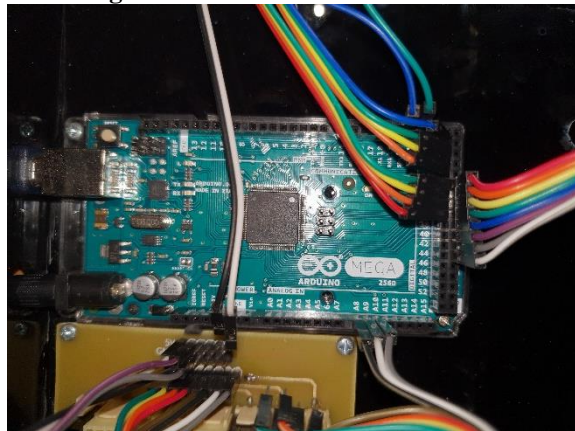
Arduino ethernet shield pada sistem ini digunakan sebagai ekstensi dari arduino mega 2560 untuk dapat melakukan koneksi terhadap internet melalui kabel ethernet.





**Gambar 3.14.** Tata Letak Ethernet Shield pada Arduino Mega 2560.

### 3.2.8. Arduino Mega 2560



**Gambar 3.15.** Desain Peletakan Arduino Mega 2560 pada *Control Unit*.

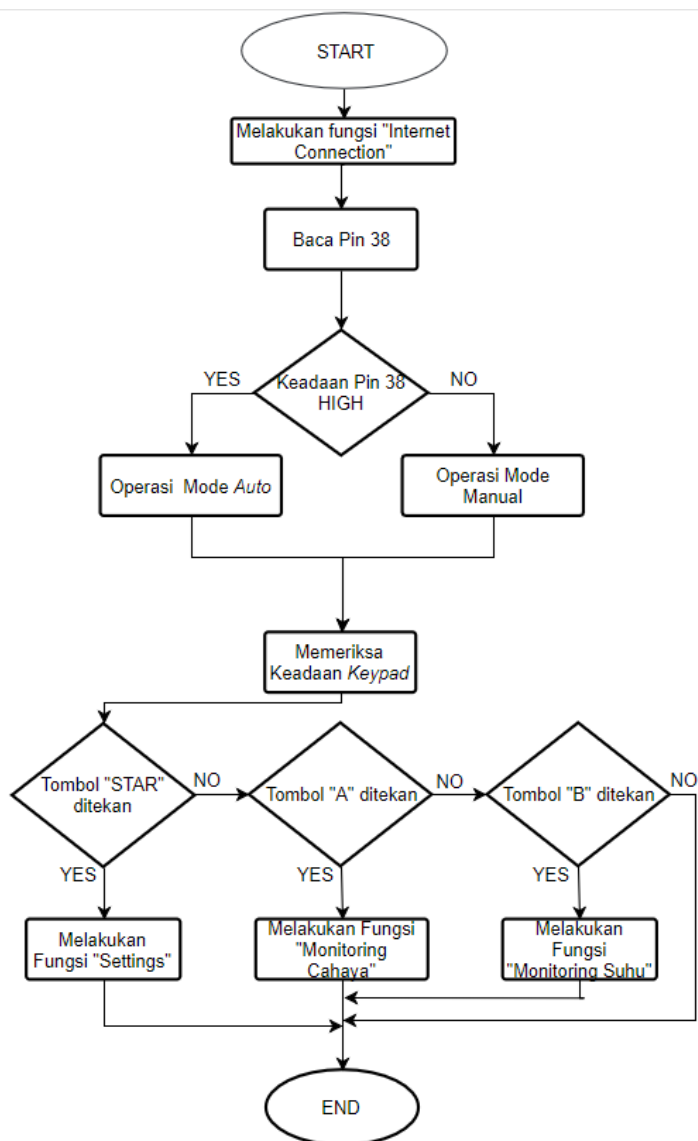
Arduino pada sistem ini adalah sebagai kontroler yang memiliki fungsi untuk mengolah data dari sensor, mengatur pola kerja dari relay yang adalah aktuatur dalam sistem ini, dan menghubungkan *control unit* ke internet melalui hubungan *ethernet*. Arduino Mega digunakan sebagai *controller* pada sistem ini atas alasan banyaknya

ketersediaan pin I/O digital pada *board*, memiliki kompatibilitas dengan arduino ethernet shield dan *processor* yang relatif cukup cepat untuk mengolah data yang dibutuhkan pada sistem ini. Pin pada arduino mega yang digunakan pada sistem ini dapat dilihat pada bagian lampiran.

### 3.2.9. Perancangan Program Utama

Pada *loop* utama program arduino di sistem ini, dilakukan beberapa fungsi. Fungsi utama yang dilakukan pada bagian ini adalah untuk menentukan mode yang beroperasi pada sistem. Mode yang beroperasi pada sistem ini ditentukan oleh keadaan pada pin *digital* D38. Bila pin D38 bernilai logika “*HIGH*”, maka sistem akan menjalankan mode *auto*, sebaliknya bila pin D38 bernilai logika “*LOW*”, maka sistem akan menjalankan mode manual. Keadaan pada pin D38 (“*HIGH*” atau “*LOW*”) ditentukan oleh suatu nilai pada *web server*. Untuk itu, pada *loop* utama ini juga dilakukan koneksi dengan internet.

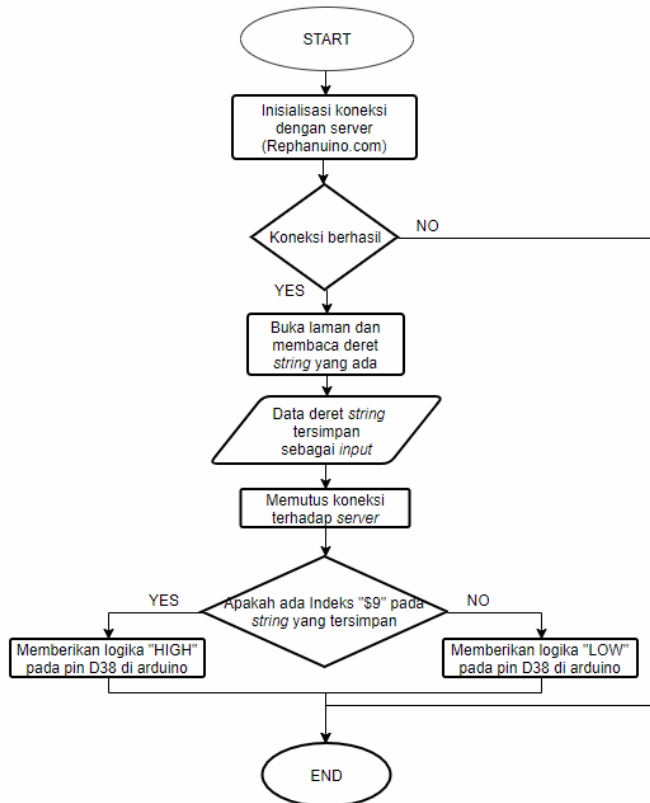
Fungsi lain dari program utama ini adalah melakukan pemeriksaan terhadap tombol pada *membrane keypad*. Bila tombol “*STAR*” pada *keypad* ditekan maka program akan menjalankan fungsi “*settings*”, bila tombol “*A*” pada *keypad* ditekan maka program akan menjalankan fungsi “*monitoring cahaya*”, dan bila tombol “*B*” pada *keypad* ditekan, maka program akan menjalankan fungsi “*monitoring suhu*”. Fungsi “*monitoring cahaya*” berfungsi untuk melihat nilai yang sedang dibaca oleh sensor cahaya dalam bentuk persentase, kemudian fungsi “*monitoring suhu*” berfungsi untuk melihat nilai yang sedang dibaca oleh sensor suhu dalam satuan celsius. Pola kerja dari program *loop* utama sistem ini diilustrasikan oleh *flowchart* yang dipaparkan pada gambar 3.16 berikut ini:



**Gambar 3.16.** *Flowchart Program Arduino Utama.*

### 3.2.10. Perancangan Program *Internet Connection*

Pada perancangan program arduino untuk sistem ini, ada segmen khusus yang menjalankan fungsi untuk melakukan koneksi internet antara arduino dengan *web server* yang telah dibuat (Rephanuino.com). Pada segmen ini juga dilakukan pengaturan keadaan pada pin 38 arduino mega, yang menjadi indikator mode operasi pada sistem *building automation* ini. Pada aktivitas koneksi ini, arduino berperan sebagai *client* yang mengambil data dari *server*. Pola kerja program ini dapat dilihat pada gambar *flowchart* berikut.



**Gambar 3.17.** *Flowchart* Program Koneksi Internet pada Arduino.

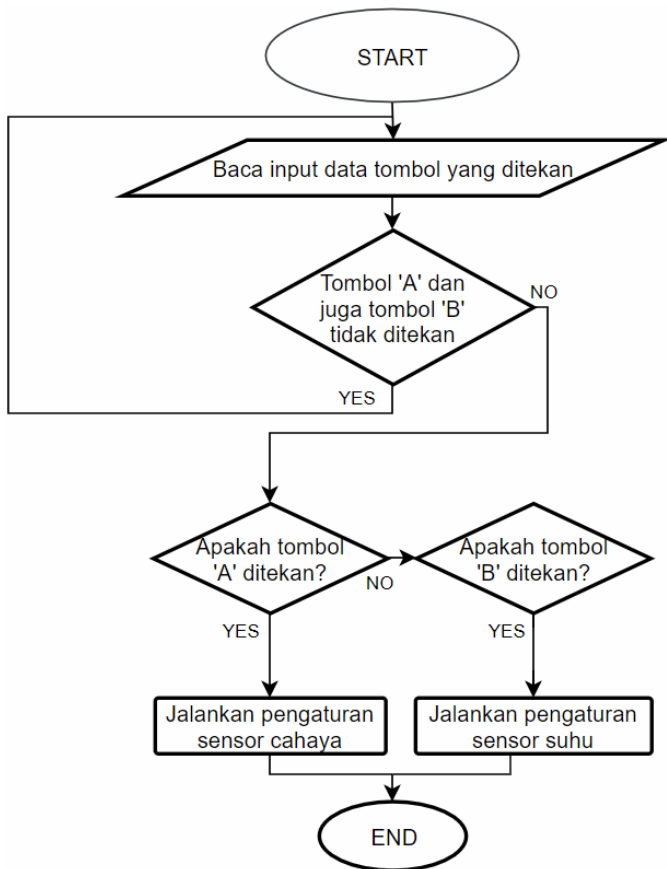
Pertama-tama setelah fungsi program ini terinisialisasi, arduino akan mencoba melakukan koneksi terhadap *server* dengan perantara dari ethernet shield. Bila koneksi ke *server* tidak berhasil dilakukan maka fungsi ini akan berakhir dan kembali ke *loop* utama. Namun, bila koneksi ke *server* berhasil dilakukan maka arduino kemudian akan membuka laman bernama <http://rephanuino.com/BuildingAutomation/viewdbUI.php> pada *server*, dan membaca deret karakter pada laman tersebut. Deret karakter yang terbaca saat itu kemudian disimpan dalam sebuah variabel dalam bentuk *string*, dan arduino kemudian memutuskan koneksi ke *web server*.

Deret karakter yang didapat pada dasarnya terdiri dari 9 bagian, di mana 9 bagian pada deret tersebut akan menjadi referensi *state* yang merepresentasikan 8 pin channel relay dan 1 pin *mode selector* pada sistem. Pada segmen ini, deret karakter yang tersimpan akan diperiksa oleh arduino, dan bila dalam deret karakter tersebut terdapat indeks “\$9” di dalamnya, maka arduino akan memberikan nilai logika “HIGH” pada pin D38. Sebaliknya bila arduino menemukan indeks “\$i” pada deret karakter tersebut, maka pin D38 akan diberikan logika “LOW”. Setelah itu, segmen program ini akan berakhir dan kembali ke *loop* utama.

### 3.2.11. Perancangan Program *Settings*

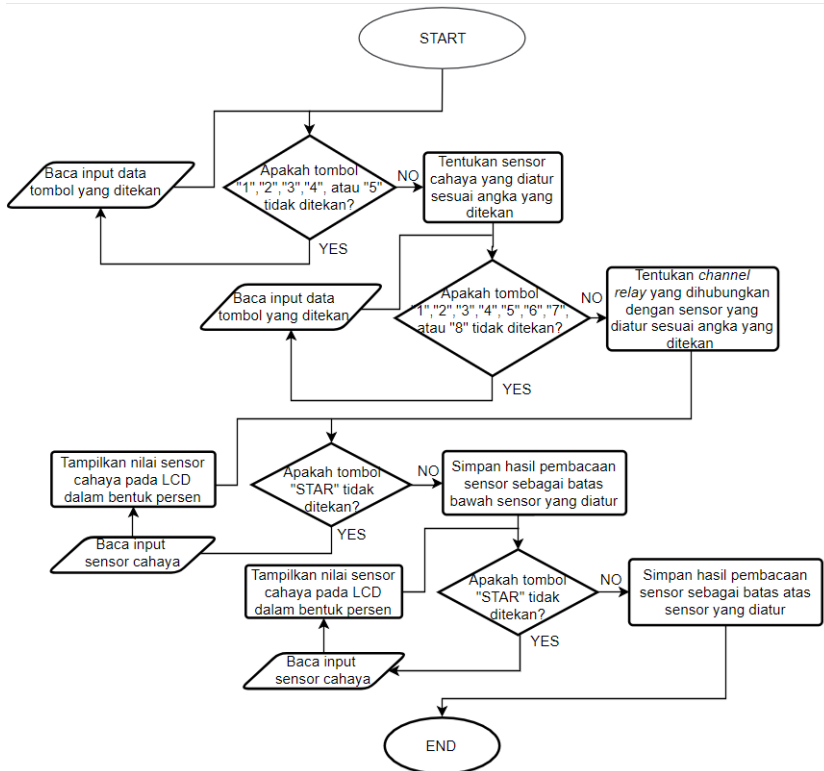
Salah satu fitur penting yang ada pada sistem *building automation* ini adalah adanya pemberian kebebasan kepada *user* dalam mengatur aktivitas pada sensor. Aktivitas pada sensor di sistem ini bekerja dengan berdasarkan referensi yaitu nilai batas atas dan batas bawah dari pembacaan sensor. Batas bawah dan batas atas ini yang kemudian akan ditentukan oleh *user* melalui fungsi “*settings*” pada program. Selain itu, fungsi *settings* pada program ini juga berperan untuk menentukan sensor pada *channel* apa saja yang akan dipakai oleh *user*, dan menghubungkannya masing-masing dengan *channel* keluaran yang terhubung dengan *relay*. Setelah sensor dan *relay* terhubung.

Fungsi *settings* ini bekerja untuk menentukan parameter-parameter pada mode operasi *auto* pada sistem, sedangkan untuk mode manual, pengaturan yang telah ditetapkan pada fungsi *settings* tidak akan mempengaruhi aktivitas dari sistem.



**Gambar 3.18.** *Flowchart Program Settings Utama.*

Pada program *settings* di sistem ini, pengaturan yang dilakukan terbagi ke dalam 2 kategori sesuai dengan jenis sensornya (cahaya atau suhu). Pada gambar 3.18 diperlihatkan *flowchart* dari program penentuan jenis sensor yang akan diatur oleh *user*. Program akan menunggu hingga *user* menekan tombol ‘A’ atau ‘B’ pada *keypad* dan menentukan fungsi pengaturan yang dilakukan. Bila tombol ‘A’ ditekan maka dilakukan pengaturan berbasis sensor cahaya, dan bila tombol ‘B’ ditekan maka dilakukan pengaturan berbasis sensor suhu.

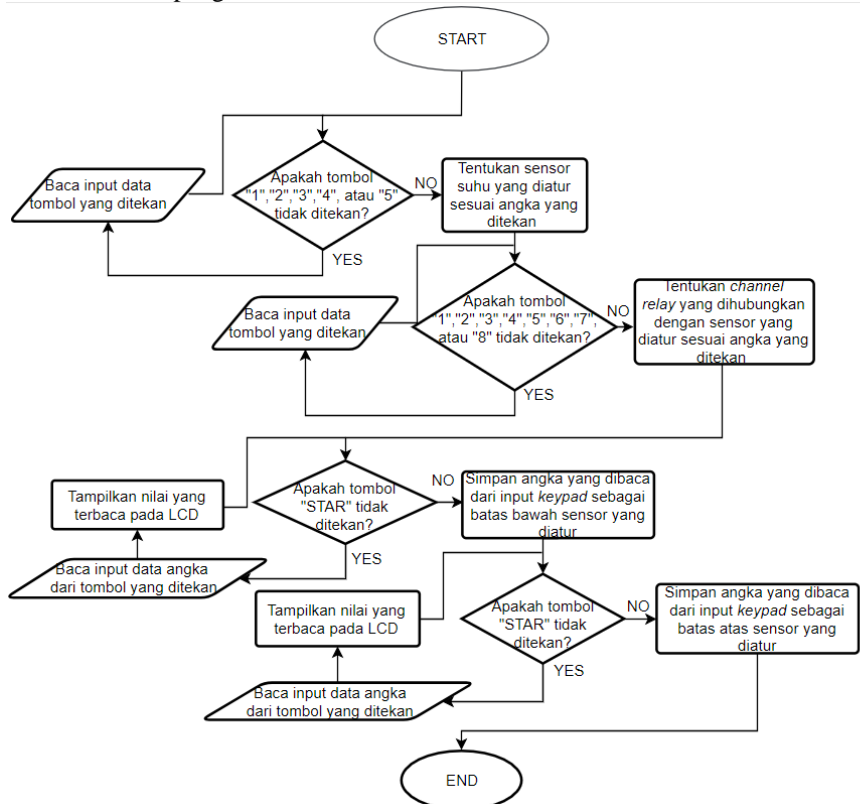


**Gambar 3.19.** Flowchart Program Settings Pengaturan Sensor Cahaya.

Bila *user* memilih pengaturan berdasarkan sensor cahaya, maka program akan memasuki bagian seperti yang diilustrasikan pada gambar 3.19. Pada bagian ini *user* pertama-tama memilih sensor cahaya yang akan diatur dengan menekan salah satu tombol angka dari 1 hingga 5. Kemudian akan diperlihatkan pilihan angka pada LCD yang merepresentasikan *channel* pada *relay* yang akan dihubungkan aktivitasnya dengan sensor yang telah ditentukan sebelumnya. Pemilihan *channel* akan dilakukan dengan menekan salah satu tombol angka pada *keypad* antara 1 hingga 8.

Kemudian setelah itu pengaturan yang dilakukan adalah untuk menentukan nilai yang menjadi batas bawah dan batas atas dari

pembacaan nilai sensor cahaya. Yang dilakukan terlebih dahulu adalah penentuan batas bawah dari sensor cahaya. LCD akan menampilkan angka pembacaan yang dilakukan sensor cahaya dalam bentuk persentase, kemudian peran *user* adalah untuk menekan tombol “STAR” pada *keypad* saat keadaan cahaya yang diinginkan telah dicapai, lalu program akan menyimpan nilai pembacaan sensor cahaya pada keadaan tersebut untuk dijadikan nilai batas bawah sensor tersebut. Setelah itu dilakukan pengaturan nilai batas atas untuk sensor tersebut dengan menggunakan metode yang sama dengan penentuan batas bawah yang dijelaskan sebelumnya, hanya saja nilai yang disimpan akan dijadikan batas atas untuk sensor. Setelah itu pengaturan untuk sensor tersebut berakhir.



**Gambar 3.20.** Flowchart Program Settings Pengaturan Sensor Suhu.



Bila pada program *settings* utama *user* memilih untuk mengatur sensor suhu, maka program akan menjalankan fungsi seperti yang ditunjukkan pada *flowchart* di gambar 3.20. *User* pertama-tama memilih sensor suhu yang akan diatur dengan menekan salah satu tombol angka dari 1 hingga 3. Kemudian akan diperlihatkan pilihan angka pada LCD yang merepresentasikan *channel* pada *relay* yang akan dihubungkan aktivitasnya dengan sensor yang telah ditentukan sebelumnya. Pemilihan *channel* akan dilakukan dengan menekan salah satu tombol angka pada *keypad* antara 1 hingga 8.

Kemudian dilakukan pengaturan nilai batas bawah dan batas atas untuk sensor suhu. Sama seperti pada pengaturan sensor cahaya, penentuan batas bawah dilakukan terlebih dahulu. Pada bagian ini, LCD akan menampilkan angka yang ditekan pada *keypad* oleh *user*, sebelum tombol “STAR” pada *keypad* ditekan. Saat tombol “STAR” ditekan oleh *user*, maka angka yang terakhir tersimpan (yang ditampilkan pada LCD), akan disimpan dalam suatu variabel dan dijadikan batas bawah untuk pembacaan oleh sensor suhu yang digunakan. Selanjutnya dilakukan penentuan batas atas untuk sensor suhu yang diatur. Metode yang digunakan sama seperti saat menentukan batas bawah sensor suhu, hanya saja angka yang disimpan pada bagian ini akan dijadikan nilai batas atas untuk sensor suhu yang diatur.

### 3.2.12. Perancangan Program Operasi *Auto-Mode*

Operasi *auto mode* pada dasarnya adalah mengaktifkan atau mematikan *channel* pada *relay module* berdasarkan *upper limit* dan *lower limit* yang telah ditentukan pada fungsi “*settings*” melalui *user interface* berupa *keypad* dan LCD. Untuk aktivitas berdasarkan sensor cahaya, dilakukan dengan program berikut:

```
if (LightValue[s[0]] <= LightBottomLimit[0])
    digitalWrite(RelayChannel[r[0]], HIGH);
else if (LightValue[s[0]] >= LightUpperLimit[0])
    digitalWrite(RelayChannel[r[0]], LOW);

if (LightValue[s[1]] <= LightBottomLimit[1])
    digitalWrite(RelayChannel[r[1]], HIGH);
else if (LightValue[s[1]] >= LightUpperLimit[1])
```

```

digitalWrite(RelayChannel[r[1]], LOW);

if (LightValue[s[2]] <= LightBottomLimit[2])
    digitalWrite(RelayChannel[r[2]], HIGH);
else if (LightValue[s[2]] >= LightUpperLimit[2])
    digitalWrite(RelayChannel[r[2]], LOW);

if (LightValue[s[3]] <= LightBottomLimit[3])
    digitalWrite(RelayChannel[r[3]], HIGH);
else if (LightValue[s[3]] >= LightUpperLimit[3])
    digitalWrite(RelayChannel[r[3]], LOW);

if (LightValue[s[4]] <= LightBottomLimit[4])
    digitalWrite(RelayChannel[r[4]], HIGH);
else if (LightValue[s[4]] >= LightUpperLimit[4])
    digitalWrite(RelayChannel[r[4]], LOW);

```

Pada program di atas bagian pertama, variabel *array* “s[0]” hingga “s[4]” merepresentasikan sensor yang digunakan. Variabel ini diisi dengan nilai integer pada saat pengaturan pada fungsi “*settings*”. Bila sensor C1 tidak digunakan, maka nilai “s[0]” akan menjadi “0”, di mana variabel “LightValue[0]” tidak didefinisikan, sehingga fungsi tersebut tidak akan memengaruhi kerja dari sistem. Namun apabila sensor C1 digunakan, maka nilai “s[0]” akan menjadi “1”, dan variabel “LightValue[1]” telah didefinisikan sebagai nilai yang didapat dari sensor C1, sehingga fungsi pertama berjalan.

Bila sensor digunakan, kemudian nilai dari “LightValue[1]” akan dibandingkan dengan variabel “LightBottomLimit[0]” dan “LightUpperLimit[0]”, yang merupakan nilai dari batas bawah dan batas atas pada sensor C1 yang ditentukan pada saat *user* melakukan pengaturan pada fungsi “*settings*”. Bila “LightValue[1]” bernilai lebih kecil dari “LightBottomLimit[0]” maka *relay* pada *channel* yang ditentukan akan aktif, kemudian bila bernilai lebih besar dari “LightUpperLimit[0]”, maka *relay* pada *channel* yang ditentukan akan mati. Begitu pula dengan sensor cahaya C2 hingga C5.

Kemudian untuk aktivitas berdasarkan sensor suhu mengikuti program berikut:

```

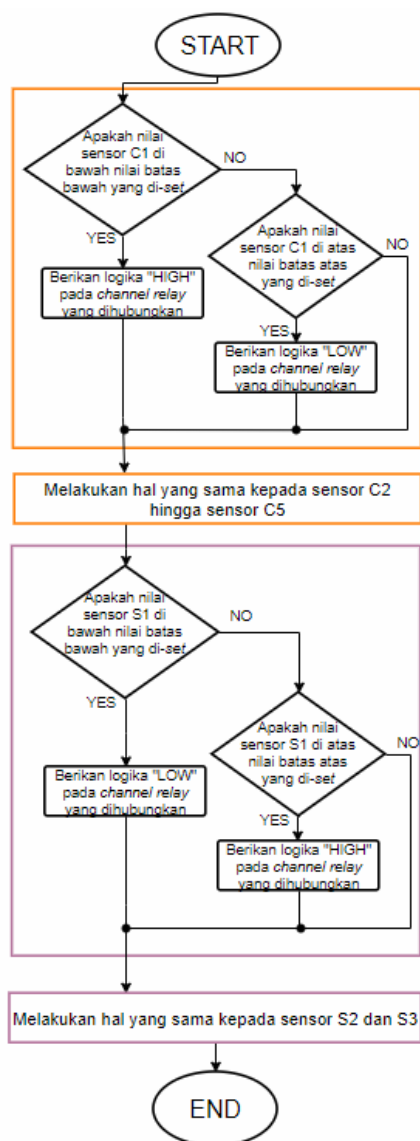
if (TempValue[t[0]] <= TempBottomLimit[0])
    digitalWrite(RelayChannel[r[5]], LOW);
else if (TempValue[t[0]] >= TempUpperLimit[0])
    digitalWrite(RelayChannel[r[5]], HIGH);

if (TempValue[t[1]] <= TempBottomLimit[1])
    digitalWrite(RelayChannel[r[6]], LOW);
else if (TempValue[t[1]] >= TempUpperLimit[1])
    digitalWrite(RelayChannel[r[6]], HIGH);

if (TempValue[t[2]] <= TempBottomLimit[2])
    digitalWrite(RelayChannel[r[7]], LOW);
else if (TempValue[t[2]] >= TempUpperLimit[2])
    digitalWrite(RelayChannel[r[7]], HIGH);

```

Untuk program di atas, cara kerjanya mirip dengan bagian program yang sebelumnya untuk sensor cahaya, hanya saja fungsinya dibalik, di mana *relay* akan aktif saat nilai sensor di atas batas atas, dan akan mati saat nilai sensor di bawah batas bawah yang ditentukan.



**Gambar 3.21.** Flowchart Program Operasi Mode Auto.

### 3.2.13. Perancangan Program *Manual-Mode*

Pada *manual mode* sistem akan bekerja sesuai dengan kondisi deret karakter yang tersimpan melalui program bagian “*internet connection*”. Pada mode ini, kondisi relay pada setiap *channel* direpresentasikan oleh karakter tersendiri yang disimpan pada saat program “*internet connection*” berjalan. Karakter untuk setiap *relay* dan kondisinya diperlihatkan pada tabel 3.3.

**Tabel 3.3.** Karakter Representasi Kondisi Relay.

<i>Channel Relay</i>	<b>Kondisi Relay</b>	
	<i>ON</i>	<i>OFF</i>
1	\$1	\$a
2	\$2	\$b
3	\$3	\$c
4	\$4	\$d
5	\$5	\$e
6	\$6	\$f
7	\$7	\$g
8	\$8	\$h

Kemudian program pada mode manual ini mengubah kondisi pada *channel relay* yang bersangkutan sesuai dengan kondisi karakter yang tersimpan dengan program berikut:

```
{
    for (int n = 0; n < 8; n++)
    {
        if (readString.indexOf(arrayon[n]) > 0)
        {
            digitalWrite(((2 * n) + 22), HIGH);
        }
    }
}
```

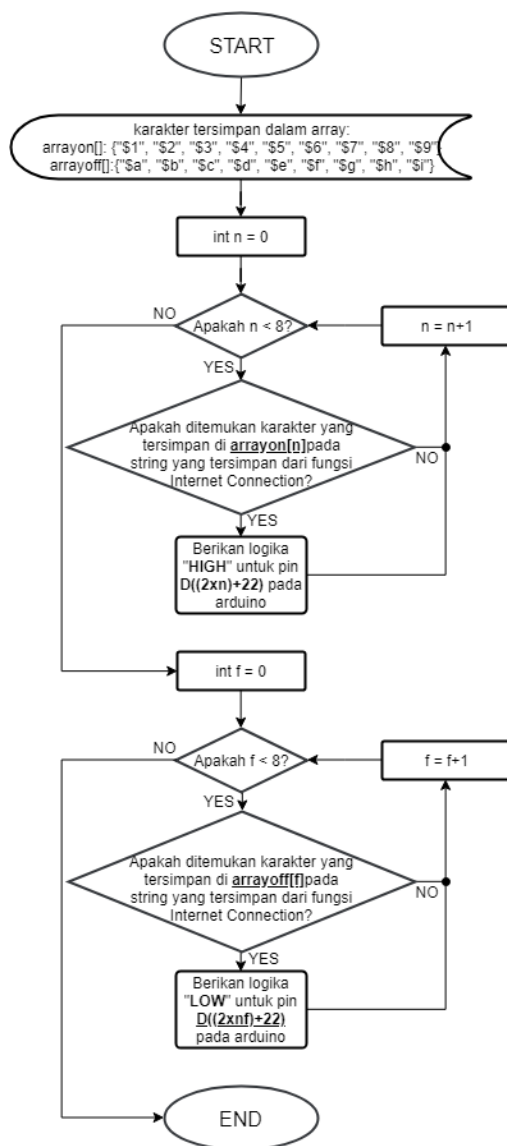
```

for (int f = 0; f < 8; f++)
{
    if (readString.indexOf(arrayoff[f]) > 0)
    {
        digitalWrite(((2 * f) + 22), LOW);
    }
}

readString = ""; //clear readString variable
}

```

Pada potongan program di atas, dilakukan fungsi for loop untuk melakukan pemeriksaan terhadap karakter yang merepresentasikan kondisi *channel relay* satu per satu. Fungsi “indexOf()” berfungsi untuk memeriksa karakter yang tersimpan pada variabel “readString”. “readString” merupakan variabel yang menyimpan karakter yang terbaca pada *website* dalam bentuk *string*. Bila karakter yang sedang dicari ada pada variabel “readString”, maka arduino akan mengaktifkan atau mematikan *channel relay* yang bersangkutan sesuai karakter yang ditemui pada “readString”.

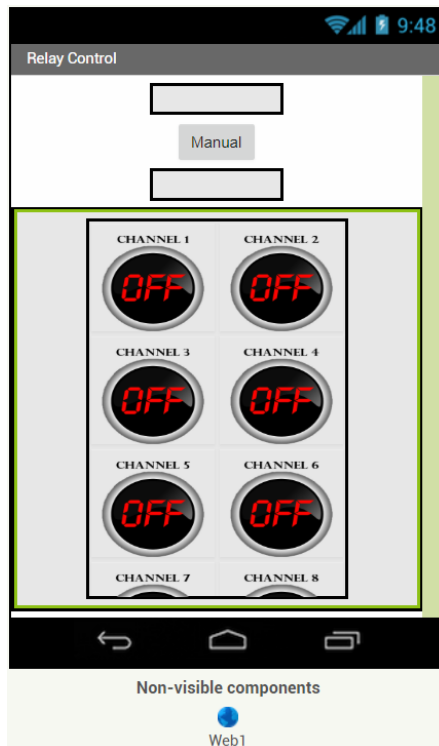


**Gambar 3.22.** Flowchart Program Operasi Mode Manual.

### 3.4. Perancangan *Command Center*

*Command center* pada sistem ini adalah tempat di mana *user* dapat memberikan perintah langsung terhadap sistem *building automation* ini. *Command center* pada sistem ini berupa aplikasi *smartphone* android, yang mana fungsinya menentukan mode yang dijalankan oleh sistem, dan mengendalikan sistem secara langsung pada saat sistem dalam keadaan operasi manual.

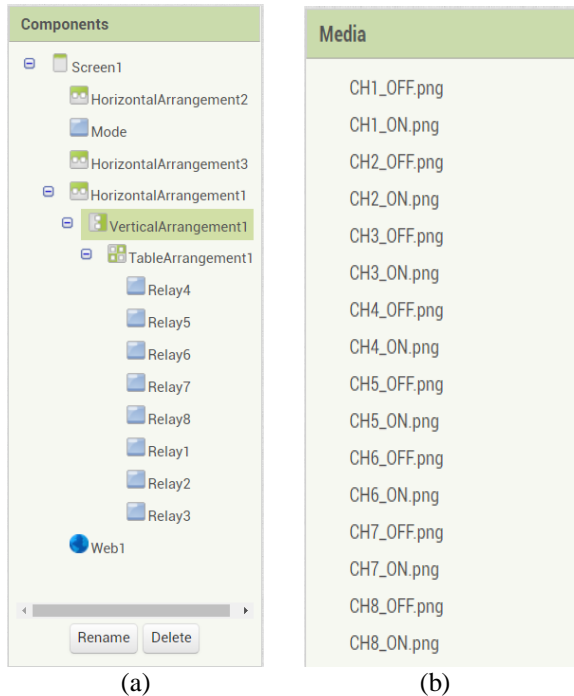
#### 3.3.1. Perancangan *Design Interface* dan Komponen Aplikasi



**Gambar 3.23.** Desain *Interface* pada Aplikasi.

Interface pada aplikasi android yang telah dibuat memiliki satu halaman, dikarenakan fungsinya yang tidak memerlukan banyak halaman.





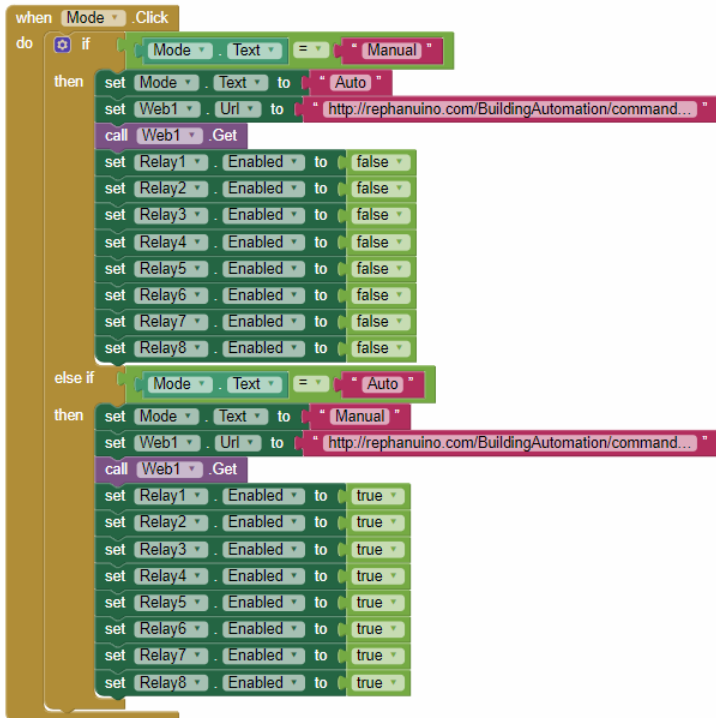
**Gambar 3.24.** (a)Komponen yang Digunakan pada Aplikasi;  
(b)Gambar yang Digunakan pada Tombol.

Dari gambar 3.23 dan gambar 3.24 terlihat *interface* pada aplikasi yang dirancang untuk sistem ini beserta komponen yang digunakan di dalamnya. Komponen yang ada pada aplikasi android ini terdiri dari 9 tombol, satu di antaranya adalah tombol “Mode” yang akan mengubah mode operasi pada sistem saat ditekan, lalu 8 tombol yang lainnya adalah tombol “Relay1” hingga “Relay8” yang akan mengubah kondisi *channel* pada *relay board* yang bersangkutan saat ditekan. File gambar pada kolom “media” digunakan sebagai indikator tombol “Relay1” hingga “Relay8”, yang akan berubah-ubah secara bergantian menampilkan gambar “ON” dan “OFF” saat tombol “Relay” ditekan.

Ada pula komponen yang tak terlihat pada aplikasi android ini, komponen tersebut adalah komponen “web viewer”, di mana

tugasnya adalah untuk mengakses dan melihat laman dari *website* tertentu. Pada sistem *building automation* ini, fungsi ini digunakan agar aplikasi dapat mengakses laman pada *website* dan mengubah kondisi dari suatu nilai pada *database* dalam *web server*. Hal ini dapat dilakukan dengan cara mengakses URL tertentu yang berkorelasi dengan pemrograman yang telah dibuat pada *web server*.

### 3.3.2. Perancangan Program Blok Penentuan Mode Operasi



**Gambar 3.25.** Program Penentuan Mode Operasi.

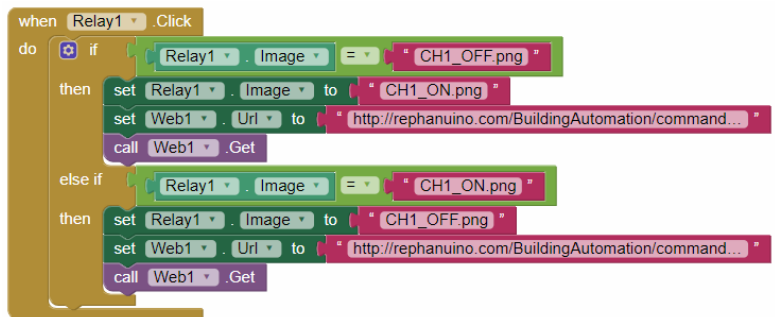
Pada gambar 3.25 terlihat *block programming* yang mengatur aktivitas dari aplikasi android saat tombol “Mode” pada *interface* (gambar 3.23 dan 3.24) ditekan oleh *user*. Jika tombol “Mode” ditekan saat tombol tersebut memperlihatkan tulisan “Manual”, maka

tombol akan mengubah tulisannya menjadi “Auto” dan kemudian mengakses URL berikut ini: “http://rephanuino.com/BuildingAutomation/command9.php?KONDISI9=9.”

Yang dilakukan saat mengakses URL tersebut adalah membuka folder “BuildingAutomation” yang ada pada website “http://rephanuino.com” di dalam web server, lalu membuka file “command9.php” yang adalah file program yang ada pada web server. Kemudian memberikan nilai pada variabel “KONDISI9” yang ada di dalam file “command9.php” menjadi “9”. Setelah mengakses URL tersebut, tombol “Relay1” hingga “Relay8” pada aplikasi akan dinonaktifkan sehingga tombol tersebut tidak akan melakukan respon saat ditekan oleh *user*, hal ini dilakukan untuk mencegah penumpukkan perintah pada aplikasi android.

Kemudian bila tombol “Mode” ditekan saat menampilkan tulisan “Auto” maka tombol akan mengubah tulisannya menjadi “Manual”, dan mengakses URL yang sama seperti sebelumnya, hanya saja variabel “KONDISI9” diisi dengan karakter “i”, setelah itu tombol “Relay1” hingga “Relay8” diaktifkan dan *user* dapat menekan tombol tersebut untuk menjalankan suatu perintah.

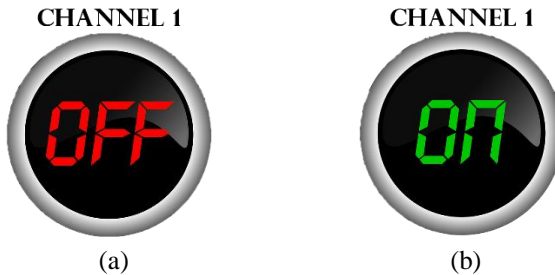
### 3.3.3. Perancangan Program Blok Kendali Operasi Manual



**Gambar 3.26.** Program Kendali Manual.

Aplikasi android pada sistem ini selain berfungsi untuk mengatur mode operasi pada sistem, juga melakukan kendali terhadap *channel* pada *relay* pada saat sistem menjalankan operasi manual. Kendali yang dilakukan oleh aplikasi android ini bekerja berdasarkan penekanan tombol pada *interface*. Pada gambar 3.26 diperlihatkan

blok program yang mengatur aktivitas yang dilakukan saat tombol “Relay1” ditekan.



**Gambar 3.27.** (a) gambar file “CH1\_OFF.png”; (b) gambar file “CH1\_ON.png”.

Bila tombol “Relay1” ditekan saat tombol menampilkan gambar “CH1\_OFF.png” maka tombol akan mengubah tampilannya menjadi gambar “CH1\_ON.png”, setelah itu dilakukan pengaksesan URL: “<http://rephanuino.com/BuildingAutomation/command1.php?KONDISI1=1>”, yang fungsinya mirip dengan yang telah dijelaskan pada bagian perancangan program blok penentuan mode operasi, hanya saja file yang dibuka di sini adalah “command1.php”, dan variabel yang diubah adalah “KONDISI1”. Dengan cara yang sama dapat diketahui pula cara kerja dari program blok selanjutnya yang terlihat pada gambar 3.26. Tombol “Relay2” hingga “Relay8” bekerja dengan cara yang sama dengan tombol “Relay1”, hanya saja file .php yang diakses dan variabel yang diubah berbeda sesuai dengan *channel* yang dikontrol oleh tombol tersebut.

### 3.5. Perancangan Web Server

*Web server* pada sistem ini adalah tempat di mana nilai yang akan menjadi referensi yang menentukan keadaan dari *control unit* disimpan. *Web server* pada sistem ini dibuat menggunakan cara *hosting* di mana *developer* mengambil sebagian kapasitas pada komputer *server* yang sudah ada untuk membuat *website* dengan *domain* sendiri. *Domain* yang digunakan pada *web server* ini menggunakan nama “<http://rephanuino.com>”.

### 3.3.4. Perancangan Database pada Web Server

Pada *web server* yang digunakan oleh domain “http://rephanuino.com”, dibuat tabel yang dimasukkan ke dalam *database server*. Tabel yang dibuat tersebut merupakan tabel kondisi, di mana tabel ini menyimpan karakter yang akan menjadi referensi yang dibaca oleh *control unit* pada sistem.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 ID	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	2 LED1	varchar(20)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 LED2	varchar(10)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 LED3	varchar(10)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5 LED4	varchar(10)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	6 LED5	varchar(10)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	7 LED6	varchar(10)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	8 LED7	varchar(10)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	9 LED8	varchar(10)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	10 LED9	varchar(10)	latin1_swedish_ci		No	None			Change  Drop  More

**Gambar 3.28.** Struktur Tabel Kondisi pada *Database Web Server*.

Pada gambar 3.28 terlihat struktur dari tabel kondisi yang telah dibuat. Tabel ini memiliki 10 kolom yang menyimpan nilai di dalamnya. Kolom pertama adalah kolom ID, kemudian kolom ke-2 hingga kolom ke-10 merupakan kolom yang menyimpan data variabel dalam bentuk karakter (varchar), jenis karakter yang disimpan pada setiap kolom dari kolom ke-2 hingga kolom ke-10 mengikuti bahasa *encoding* “latin1\_swedish\_ci”.

Nama dari setiap kolom dari kolom ke-2 hingga kolom ke-10 adalah LED1, LED2, LED3, LED4, LED5, LED6, LED7, LED8, dan LED9. Tiap kolom tersebut menyimpan sebuah variabel, variabel - variabel ini lah yang akan menyimpan karakter yang kemudian akan diubah-ubah melalui perintah dari *command center*.

<div><div><div>←</div><div>→</div></div><div><div></div><div></div></div></div>				ID	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED8	LED9
<div><div></div><div></div></div>	<div><div>Edit</div><div></div></div>	<div><div>Copy</div><div></div></div>	<div><div>Delete</div><div></div></div>	0	1	b	c	4	5	f	g	8	i

**Gambar 3.29.** Tampilan Tabel Kondisi yang Menyimpan Nilai Karakter Dalamnya.

Gambar 3.29 memperlihatkan kondisi pada tabel kondisi dengan karakter yang tersimpan di dalamnya melalui suatu variabel. Bila *command center* memberikan suatu perintah sesuai fungsinya, maka karakter yang tersimpan dalam kolom – kolom di gambar 3.29 akan berubah sesuai dengan perintahnya.

### 3.3.5. Perancangan Program pada Web Server

Pada *web server* dibuat beberapa file yang berisi program dalam bahasa *hypertext preprocessor* (dengan ekstensi .php) yang disimpan dalam *folder* bernama “BuildingAutomation”. File berekstensi .php ini memiliki fungsi masing – masing yang berbeda satu sama lainnya.

/public\_html/BuildingAutomation/connect.php

```
1 <?php
2 $db_host = 'mysql.hostinger.co.id';
3 $db_user = 'u121424105_root';
4 $db_pass = '12345678';
5 $db_name = 'u121424105_coba';
6
7 $connect = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
8 if (!$connect) {
9     die ('Failed to connect to your database: '.mysqli_connect_error());
10 }
11 ?>
```

**Gambar 3.30.** Isi Program dalam File “connect.php”.

Pada *file* ini dilakukan fungsi koneksi, yang mana koneksi yang dilakukan adalah dengan sebuah *database* yang telah tersimpan dalam *web server*. *Database* yang akan dihubungkan ini merupakan *database* yang berisi tabel kondisi yang telah dijelaskan pada bagian sebelumnya. Program yang diperlihatkan pada gambar 3.30 koneksi dilakukan dengan memanggil fungsi “*mysqli\_connect()*” diikuti dengan parameter yang dibutuhkan, yaitu: alamat *host*, *user*, *password*, dan nama *database* yang ingin dihubungkan.

/public\_html/BuildingAutomation/command1.php

```
1 k?php
2 $db_host = 'mysql.hostinger.co.id';
3 $db_user = 'u121424105_root';
4 $db_pass = '12345678';
5 $db_name = 'u121424105_coba';
6
7 $connect = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
8 if (!$connect) {
9     die ('Failed to connect to your database: '.mysqli_connect_error());
10 }
11
12 $led1 = $_GET['KONDISI1'];
13 $query = mysqli_query($connect, "UPDATE tb_kondisi SET LED1='$led1'");
14
15
16 ?>
17
18
```

**Gambar 3.31.** Isi Program dalam File “command1.php”.

Setelah itu, pada gambar 3.31 diperlihatkan kode program yang muncul saat file “command1.php” dibuka. Program pada file “command1.php” membuat variabel “\$led1” yang mana menyimpan nilai dari apapun yang diberikan kepada tulisan “KONDISI1”. Nilai dari “KONDISI1” diubah dengan cara menuliskan “KONDISI1=” pada URL diikuti dengan karakter apapun yang ingin diisikan ke dalamnya. Setelah itu fungsi “\$connect” dijalankan dan tabel kondisi pada database yang dihubungkan diperbarui. Kolom bernama “LED1” pada tabel kondisi sekarang akan menyimpan nilai dari variabel “\$led1” yang telah diubah sebelumnya. Dengan cara ini, kolom pada tabel kondisi dapat diubah dengan mengakses URL tertentu. Sebagai contoh pada kasus ini, bila nilai pada kolom “LED1” di tabel kondisi ingin diisikan karakter “A”, maka URL Akses URL: “http://rephanuino.com/BuildingAutomation/command1.php?KONDISI1=A”.

Pada folder “BuildingAutomation” di web server ada 9 file .php (termasuk “command1.php”) yang melakukan fungsi seperti pada gambar 3.31. Setiap file tersebut berfungsi untuk mengubah nilai pada kolom yang berbeda-beda dalam tabel kondisi. Pada sistem keseluruhan, nilai pada kolom ini lah yang menjadi referensi bekerjanya sistem melalui perintah *command center*.

/public\_html/BuildingAutomation/viewdbUI.php

```
1 <?php
2 $db_host = 'mysql.hostinger.co.id';
3 $db_user = 'u121424105_root';
4 $db_pass = '12345678';
5 $db_name = 'u121424105_coba';
6
7 $connect = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
8 if (!$connect) {
9     die ('Failed to connect to your database: ' . mysqli_connect_error());
10 }
11
12 $query1 = mysqli_query($connect,"SELECT * FROM tb_kondisi");
13 while($result = mysqli_fetch_assoc($query1))
14 {
15     $led1 = $result['LED1'];
16     $led2 = $result['LED2'];
17     $led3 = $result['LED3'];
18     $led4 = $result['LED4'];
19     $led5 = $result['LED5'];
20     $led6 = $result['LED6'];
21     $led7 = $result['LED7'];
22     $led8 = $result['LED8'];
23     $led9 = $result['LED9'];
24 }
25 echo "$"; echo $led1; echo "$"; echo $led2; echo "$"; echo $led3; echo "$"; echo $led4; echo "$"; echo $led5; echo
    "$"; echo $led6;
```

**Gambar 3.32.** Isi Program dalam File “viewdbUI.php”.

Gambar 3.32 menunjukkan isi program dari file “viewdbUI.php”. Program ini memiliki fungsi untuk menampilkan karakter pada laman *webpage*. Hal pertama yang dilakukan pada program ini adalah menghubungkan tabel kondisi pada *database server*, kemudian mengambil menyimpan nilai yang ada pada kolom “LED1”, “LED2”, “LED3”, “LED4”, “LED5”, “LED6”, “LED7”, “LED8”, dan “LED9” dalam variabel “\$led1”, “\$led2”, “\$led3”, “\$led4”, “\$led5”, “\$led6”, “\$led7”, “\$led8”, “\$led9”.

Setelah nilai tersimpan, program ini akan menampilkan setiap karakter yang telah tersimpan dalam variabel tersebut, dan sebelum setiap karakter ditampilkan karakter “\$” sebagai indikator data agar mudah dibedakan dengan karakter lain. Proses penampilan karakter ini dilakukan dengan fungsi “echo” pada bahasa PHP. Rangkaian karakter “\$” dan karakter yang tersimpan pada variabel kemudian membentuk sebuah deret karakter yang berjajar dan ditampilkan pada laman *website* saat file ini diakses. Laman yang ditampilkan oleh file ini yang kemudian akan dibaca oleh arduino melalui ethernet shield, lalu dijadikan referensi perintah dan melakukan aktivitas sesuai dengan deret karakter yang dibaca pada laman tersebut.



## BAB IV

### PENGUJIAN, HASIL, DAN ANALISIS

Pada bab ini dibahas tentang realisasi dari sistem dan pengujian - pengujian yang dilakukan terhadap komponen-komponen pada sistem yang memiliki fungsi - fungsi tersendiri yang spesifik dalam sistem. Selain itu juga dilakukan pengujian terhadap fungsionalitas dari sistem secara keseluruhan serta efektivitas dari alat terhadap fungsi penghematan energi listrik.



**Gambar 4.1.** Realisasi Desain Fisik dari *Control Unit*.

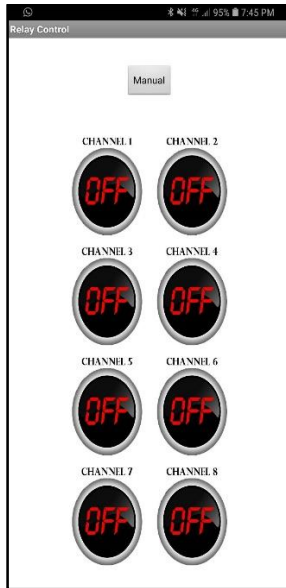
Pada gambar 4.1 diperlihatkan realisasi desain fisik *control unit* pada sistem. *Case* pada *control unit* yang menjadi wadah rangkaian elektroniknya dibuat dengan bahan akrilik. Desain dilakukan per bagian secara dua dimensi menggunakan aplikasi CorelDRAW X7. Setelah desain per bagian telah selesai, maka dilakukan pemotongan akrilik sesuai dengan desain menggunakan teknologi *laser cutting*. Bagian – bagian yang telah terpotong – potong sesuai posisinya kemudian dilekatkan satu sama lain membentuk sebuah wadah seperti terlihat pada gambar 4.1 menggunakan cairan pelekat.

Pada *case* disediakan lubang untuk menguhungkan *control unit* dengan perangkat eksternal (sumber daya, *downloader*, kabel ethernet). Dan di atas *case* ditempatkan perangkat *membrane keypad* 4x4 dan LCD 20x4 sebagai *user interface*.



**Gambar 4.2.** Tata Pengaturan Rangkaian Elektronika pada *Control Unit*.

Gambar 4.2 memperlihatkan realisasi dari perancangan sistem elektronik pada *control unit*. Arduino mega 2560 yang dilengkapi ethernet shield diletakkan di tengah sebagai pusat dari sistem pada *control unit*. Arduino kemudian dihubungkan dengan *relay board* yang ditempatkan di sebelah kiri sebagai yang kemudian akan dihubungkan terhadap perangkat elektronik yang ingin dikontrol. Pada bagian kanan arduino ditempatkan *sensor dock* yang kemudian akan menghubungkan sensor – sensor yang digunakan pada sistem kepada arduino. Terakhir, arduino dihubungkan dengan perangkat *user interface* berupa *membrane keypad* 4x4 dan LCD 20x4 yang ditempatkan di tutup bagian langit-langit pada *case* yang telah dirancang.



**Gambar 4.3.** Realisasi Tampilan Aplikasi Android sebagai *Command Center*.

Gambar 4.3 memperlihatkan tampilan dari aplikasi android yang telah dibuat pada sebuah *smartphone* android. Aplikasi ini memiliki total 9 tombol yang mana 1 tombol di antaranya adalah tombol yang mengatur mode operasi sistem, dan 8 tombol lainnya untuk memberikan perintah pada operasi manual sistem.

#### **4.1. Rekapitulasi Hasil Kuesioner**

Pada bagian ini dibahas hasil dari kuesioner yang telah disebarakan melalui media *Google Forms* kepada masyarakat yang berusia di atas 18 tahun. Kuesioner yang disebar telah diisi oleh 109 orang responden.

##### **4.1.1. Rekapitulasi Respon Terhadap Pertanyaan Tentang Kebiasaan Masyarakat**

**Tabel 4.1.** Respon Pertanyaan Tentang Kebiasaan Responden.

No.	Pertanyaan	Hasil
1.	Apakah dalam keseharian anda secara rutin menggunakan peralatan elektronik?	99.1% menjawab “Ya”
		0.9% menjawab “Tidak”
2.	Peralatan elektronik apa yang sering anda gunakan/nyalakan di rumah atau tempat anda beraktivitas?	92.7% mengisi “Lampu”
		64.2% mengisi “AC”
		55% mengisi “Kipas angin”
		70.6% mengisi “komputer”
		38.5% mengisi “Televisi”
3.	Apakah anda sering lupa mematikan peralatan elektronik yang sudah tidak dipakai di rumah atau tempat anda beraktivitas?	9.2% menjawab “sering lupa”
		56% menjawab “terkadang lupa”
		34.9% menjawab “tidak pernah lupa”
4.	Apakah alat elektronik yang sedang tidak anda gunakan biasanya tetap terhubung dengan sumber listrik? (termasuk peralatan elektronik yang sudah dalam keadaan mati)	84.4% menjawab “Ya”
		15.6% menjawab “Tidak”

Dari hasil yang didapat dan dapat dilihat pada tabel 4.1, disimpulkan bahwa hampir semua responden rutin menggunakan peralatan elektronik setiap harinya, dan lampu, kipas angin, dan AC menjadi peralatan yang relatif banyak digunakan. Terlihat pula bahwa lebih dari 50% responden terkadang atau sering lupa mematikan peralatan elektronik yang digunakan saat tidak digunakan. Sehingga alat yang dapat mengendalikan alat elektronik, khususnya lampu dan AC (atau kipas angin), secara otomatis dan jarak jauh dibutuhkan oleh masyarakat. Mayoritas dari responden juga tidak mencabut peralatan elektronik yang tidak digunakan dari sumber listrik, yang artinya fenomena “vampir listrik” dialami oleh sebagian besar masyarakat.

#### **4.1.2. Rekapitulasi Respon Terhadap Pertanyaan Tentang Ketersediaan Sarana Pendukung Sistem Otomasi Bangunan**

**Tabel 4.2.** Respon Pertanyaan Tentang Ketersediaan Sarana Responden.

No.	Pertanyaan	Hasil
1.	Apakah anda menggunakan smartphone dalam menjalani aktivitas sehari-hari?	99.1% menjawab “Ya”
		0.9% menjawab “Tidak”
2.	Apakah anda memiliki akses internet di sebagian besar waktu keseharian anda?	92.7% menjawab “Ya”
		7.3% menjawab “Tidak”
3.	Apakah anda merasa nyaman dalam mengoperasikan smartphone dan internet?	96.3% menjawab “Ya”
		3.7% menjawab “Tidak”

Dari hasil yang didapatkan dari responden yang terlihat pada tabel 4.2, disimpulkan bahwa hampir semua responden telah memenuhi sarana pendukung yang dibutuhkan untuk dapat menggunakan modul sistem otomasi bangunan yang dirancang pada penelitian ini. Sehingga modul otomasi bangunan pada penelitian ini dapat digunakan oleh kebanyakan masyarakat umum.

#### **4.1.3. Rekapitulasi Respon Terhadap Pertanyaan Tentang Kecenderungan Masyarakat Terhadap Teknologi Otomasi Bangunan**

Dari hasil yang didapatkan dan tertera pada tabel 4.3, disimpulkan bahwa hampir semua, atau mayoritas dari responden kuesioner mendukung fitur-fitur yang ditawarkan pada modul sistem otomasi bangunan yang dirancang pada penelitian ini. Dan pada pertanyaan ke-empat jawaban yang diambil dari responden berupa pendapat, dan hasilnya dapat dilihat pada bagian lampiran. Dari data pendapat yang terlampir tersebut, ada beberapa pendapat yang dapat disimpulkan bahwa sistem dalam penelitian ini diinginkan memiliki sifat *user-friendly*. Sehingga pada sistem ini diberikan pula kebebasan untuk *user* dapat mengatur aktivitas dan hubungan antara sensor dan *channel relay* melalui *user interface* berupa LCD dan *membrane keypad*.

**Tabel 4.3.** Respon Pertanyaan Tentang Kecenderungan Responden Terhadap Teknologi Otomasi Bangunan.

No.	Pertanyaan	Hasil
1.	Apakah anda ingin mengurangi beban biaya listrik anda?	94.5% menjawab “Ya”
		5.5% menjawab “Tidak”
2.	Apakah anda menginginkan peralatan elektronik di rumah atau tempat anda beraktivitas dapat bekerja secara otomatis sesuai keinginan anda?	94.5% menjawab “Ya”
		5.5% menjawab “Tidak”
3.	Apakah anda ingin dapat mengendalikan peralatan elektronik di rumah atau tempat anda beraktivitas melalui internet dengan perantaraan aplikasi <i>smartphone</i> ?	91.7% menjawab “Ya”
		8.3% menjawab “Tidak”
4.	Apakah harapan anda terhadap perkembangan teknologi otomasi bangunan?	<b>Terlampir</b>

## 4.2. Pengujian dan Kalibrasi Sensor Cahaya

Sebelum digunakan dalam sistem sebagai salah satu input data pada *control* unit, sensor cahaya dikalibrasi terlebih dahulu agar menghasilkan nilai yang dapat terbaca dengan baik oleh *user* dan dapat merepresentasikan tinggi dan rendahnya tingkat intensitas cahaya yang diterima oleh sensor. Seperti telah dijelaskan pada bab sebelumnya, sensor cahaya ini memanfaatkan metode pembagian tegangan untuk mendapatkan nilai representasi intensitas cahaya dalam angka.

Intensitas cahaya memiliki satuan yang dapat diukur yaitu satuan lux, akan tetapi dikarenakan secara umum manusia jarang merasakan atau mengukur satuan ini, sehingga akan sulit membedakan tingkat intensitas cahaya melalui nilai lux. Maka, pada sistem ini tingkat intensitas cahaya pada sensor dibuat dalam bentuk persentase sesuai batas kemampuan pembacaan pada sensor.

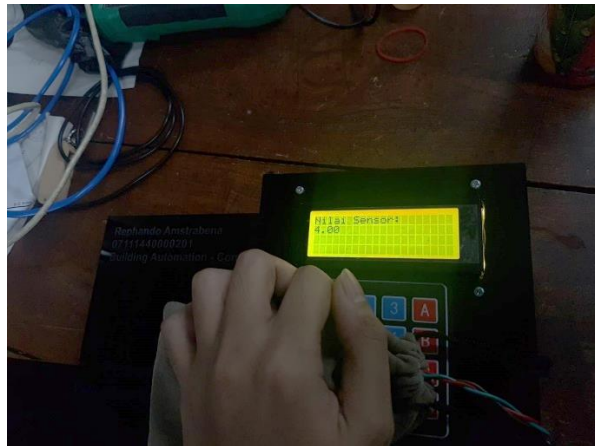
### 4.2.1. Metode Pengujian Sensor Cahaya

Untuk melakukan pengujian terhadap sensor cahaya yang telah dibuat, dilakukan pencarian nilai intensitas cahaya maksimal dan minimal yang dapat diterima oleh sensor.



**Gambar 4.4.** Pengujian Sensor Cahaya Pada Intensitas Cahaya Maksimal.

Untuk mendapatkan nilai maksimal dari sensor cahaya, digunakan cahaya *flash* dari telepon genggam dan mendekatkan sensor cahaya ke sumber cahaya tersebut hingga nilai ADC yang terbaca pada arduino tidak dapat bertambah lagi, dan mencatat nilai tersebut. Gambar 4.4 memperlihatkan proses tersebut.



**Gambar 4.5.** Pengujian Sensor Cahaya Pada Intensitas Cahaya Minimal.

Pada gambar 4.5 diperlihatkan proses dari pencarian nilai minimal sensor, yang dilakukan dengan menempatkan sensor cahaya di tempat yang sangat tertutup hingga cahaya hampir tidak ada yang masuk, lalu merapatkan celah - celah cahaya yang ada hingga nilai sensor tidak bisa menurun lagi. Nilai tersebut kemudian dicatat.

#### 4.2.2. Proses Konversi Nilai Sensor Cahaya

Ada 5 sensor yang digunakan, dengan nilai hasil yang sedikit relatif berbeda-beda, sehingga dibutuhkan standarisasi yang menyamakan variabel pengubah harga dari nilai yang dihasilkan oleh sensor.

**Tabel 4.4.** Data Nilai ADC dari Sensor Cahaya.

Sensor Ke-	Nilai Minimal (ADC)	Nilai Maksimal (ADC)
1	13	440
2	15	455
3	15	470
4	15	459
5	17	485
<b>RATA-RATA</b>	<b>15</b>	<b>461.8</b>

Pertama-tama dilakukan pengambilan data awal berupa nilai ADC, dan dicari nilai minimal dan maksimal yang dapat dibaca oleh sensor cahaya. Kemudian dari 5 sensor yang telah diambil data nilai minimal dan maksimalnya, dicari nilai rata-ratanya. Nilai rata-rata ini kemudian yang dijadikan nilai minimal dan maksimal standar untuk kelima sensor cahaya yang digunakan pada sistem.

Setelah didapatkan data yang dibutuhkan, dibuat model matematika untuk mengonversi nilai ADC yang didapat menjadi nilai persentase yang memiliki *range* pembacaan dari 0% sampai 100%. Konversi matematis tersebut dilakukan melalui persamaan:

$$\text{Persentase} = \frac{(\text{Nilai ADC} - \text{Nilai Minimal}) \times 100}{\text{Nilai Maksimal} - \text{Nilai Minimal}} \%$$

Setelah melalui persamaan tersebut, dilakukan *thresholding* yang pada nilai 0 dan 100 pada hasil konversi, sehingga saat nilai



hasil konversi melebihi 100, nilainya akan menjadi 100, dan saat nilai konversi kurang dari 0, nilainya menjadi 0. Hal ini dilakukan dengan menggunakan fungsi “IF” pada program arduino.

#### 4.2.3. Hasil Pembacaan Sensor Setelah Proses Konversi

**Tabel 4.5.** Data Nilai Persentase dari Sensor Cahaya.

Sensor Ke-	Nilai Minimal (ADC)	Nilai Maksimal (ADC)
1	0%	96%
2	0%	98%
3	0%	100%
4	0%	100%
5	0%	100%

Pada tabel 4.5 telah terlihat bahwa nilai minimal dari sensor sekarang menjadi 0%. Lalu nilai maksimal dari sensor didapat nilai yang sedikit berbeda-beda yang memiliki *range* dari 96% hingga 100%. Pada 2 sensor terdapat *error* pada nilai maksimalnya, dikarenakan perbedaan *range* nilai ADC didapatkan.

#### 4.3. Pengujian dan Sensor Suhu

Selain sensor cahaya, data *input* lain yang diolah pada sistem ini adalah *input* dari data sensor suhu. Sebelum digunakan pada sistem, perlu dilakukan kalibrasi terlebih dahulu terhadap sensor suhu yang digunakan pada sistem ini, agar sensor suhu yang digunakan kemudian menghasilkan nilai suhu dalam satuan celcius yang sesuai dengan keadaan sebenarnya. Kalibrasi ini dilakukan untuk memudahkan *user* dalam melakukan pengaturan nilai batas bawah dan batas atas nilai suhu pada fitur *settings* dalam *control unit* sistem ini.

Kalibrasi sensor suhu pada sistem ini dilakukan dengan membandingkan nilai sensor suhu pada sistem dengan nilai yang didapat pada thermometer digital. Nilai suhu dalam celcius yang dihasilkan oleh sensor idealnya adalah sama dengan nilai dihasilkan oleh thermometer digital. Karena itu, kalibrasi ini dilakukan dengan tujuan membuat keluaran sensor suhu memiliki nilai *error* sekecil-kecilnya terhadap thermometer digital yang dijadikan referensi.

#### 4.3.1. Metode Pengujian Sensor Suhu

Proses kalibrasi pada sensor suhu ini memiliki sebuah metode, yaitu dengan melakukan perbandingan antara nilai yang didapat oleh DS18B20 dengan thermometer ruangan digital. Cara yang dilakukan untuk membandingkan DS18B20 dengan thermometer digital adalah dengan memasukkan probe sensor keduanya ke dalam air berwadah, dan mengubah-ubah suhu airnya dengan memanaskan atau mendinginkannya.

Air dipilih sebagai media pengukuran karena suhu pada air menyebar dengan merata dan mudah untuk di ditempatkan pada wadah, tidak seperti udara yang suhunya kurang merata pada satu tempat dan tempat lainnya, untuk objek padat, suhunya terlalu terpusat pada satu titik, sehingga akan sulit untuk membandingkan suhu pada satu tempat dengan tempat lainnya walaupun hanya berjarak yang relatif sedikit.



**Gambar 4.6.** Pengujian Sensor Suhu Pada Air tanpa Dipanaskan atau Didinginkan.

Pengujian yang diperlihatkan pada gambar 4.6 adalah pengujian sensor pada air yang normal, tanpa dilakukan pendinginan ataupun pemanasan. Terlihat pada gambar tersebut bahwa suhu yang terbaca oleh sensor DS18B20 melalui komunikasi *one wire* relatif hampir sama nilainya dengan pembacaan oleh thermometer digital yang menjadi referensi. Sensor DS18B20 membaca nilai 25.56°C, sementara pembacaan oleh thermometer digital menunjukkan nilai 26.30°C, artinya dengan *error* yang diperoleh sebesar 2.8%.



**Gambar 4.7.** Pengujian Sensor Suhu Pada Air yang Dipanaskan.

Pada keadaan di gambar 4.7, air yang diukur telah dipanaskan, sehingga memiliki suhu yang lebih tinggi. Terlihat bahwa hasil yang diperoleh oleh sensir DS18B20 adalah  $69.31^{\circ}\text{C}$ , sedangkan yang terbaca oleh thermometer digital adalah  $70.3^{\circ}\text{C}$ . *Error* yang didapat pada percobaan ini adalah 1%.



**Gambar 4.8.** Pengujian Sensor Suhu Pada Air yang Didinginkan.

Pada gambar 4.8 diperlihatkan keadaan saat air pengukuran pada air yang telah didinginkan dengan menambahkan es pada wadah gelas. Terlihat bahwa pembacaan sensor menunjukkan angka  $3.13^{\circ}\text{C}$ ,

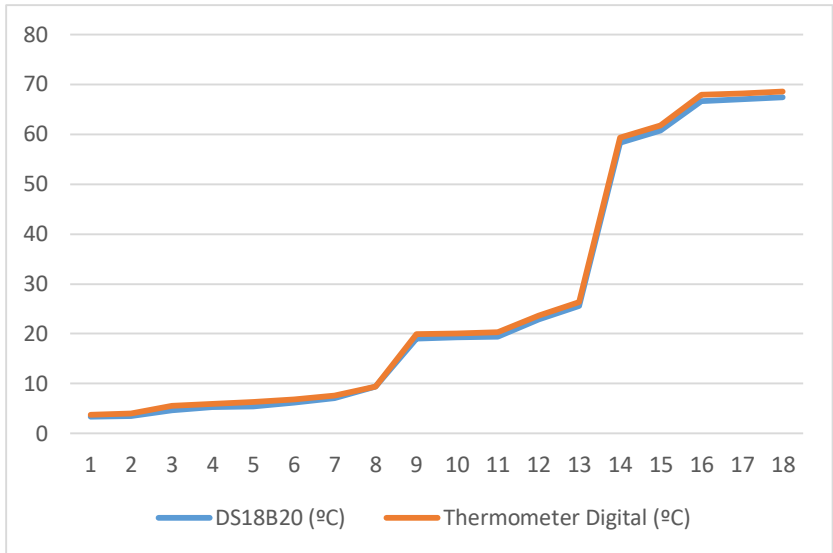
dan thermometer digital membaca nilai 3.2°C. Percobaan ini memunculkan *error* sebesar 2%.

#### 4.3.2. Hasil Pengujian Sensor Suhu

Dengan metode percobaan yang telah dijelaskan pada bagian sebelumnya, didapatkan hasil berupa tabel perbandingan antara suhu yang diperoleh oleh sensor DS18B20 dengan yang ditunjukkan oleh thermometer digital.

**Tabel 4.6.** Data Pengujian Sensor Suhu

No.	DS18B20 (°C)	Thermometer Digital (°C)	<i>Error</i>	Persen <i>Error</i> (%)
1	3,44	3,8	0,36	9,473684
2	3,5	4	0,5	12,5
3	4,63	5,6	0,97	17,32143
4	5,25	6	0,75	12,5
5	5,5	6,3	0,8	12,69841
6	6,25	6,8	0,55	8,088235
7	7,13	7,6	0,47	6,184211
8	9,38	9,4	0,02	0,212766
9	19	19,9	0,9	4,522613
10	19,25	20,1	0,85	4,228856
11	19,44	20,3	0,86	4,236453
12	22,94	23,7	0,76	3,206751
13	25,56	26,3	0,74	2,813688
14	58,25	59,3	1,05	1,770658
15	60,69	61,7	1,01	1,636953
16	66,69	67,9	1,21	1,782032
17	67,06	68,2	1,14	1,671554
18	67,44	68,5	1,06	1,547445
Rata - Rata			0,777778	5,910875



**Gambar 4.9.** Grafik Perbandingan Pembacaan Suhu Oleh DS18B20 dengan Thermometer Digital.

Dari hasil pengujian yang didapat sesuai dengan tabel 4.6, diperoleh hasil rata-rata persen *error* sebesar 5.9%. Selain itu, terlihat juga bahwa pembacaan yang diperoleh oleh sensor DS18B20 memiliki persen *error* yang relatif sama saat di kondisi suhu rendah dengan saat kondisi suhu tinggi.

Kemudian pada gambar 4.9 diilustrasikan perbandingan data pembacaan yang didapat oleh DS18B20 dengan thermometer digital, dan didapatkan *trend* yang mirip antara keduanya dan perbedaan tidak terlalu signifikan.

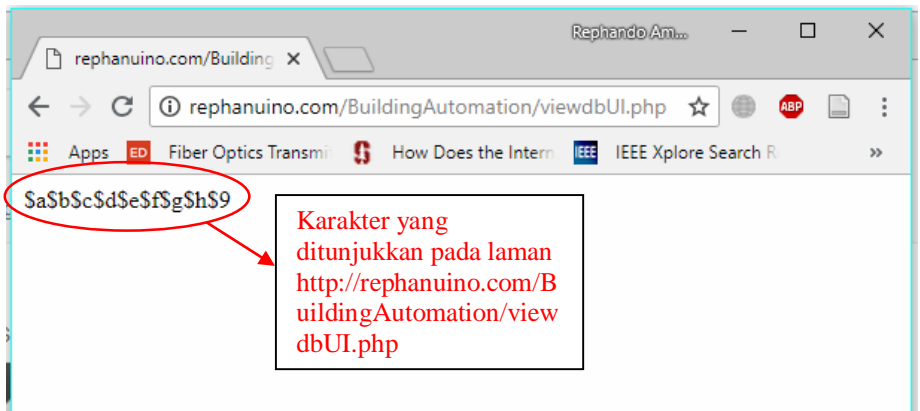
#### 4.3.3. Kesimpulan yang Diperoleh Terhadap Penggunaan Sensor

Dari data yang telah didapat atas pengujian sensor DS18B20, kemudian diambil kesimpulan setelah memasukkan variabel pertimbangan dari sisi objek pengaplikasian sensor dan penggunaan oleh *user*, maka diambil kesimpulan bahwa tidak perlu adanya proses pengalibrasian lebih lanjut. Dengan rata – rata nilai persen *error* yang telah didapat, dianggap sudah cukup akurat untuk pengaplikasiannya pada sistem *building automation* ini dan pembacaannya yang sudah

relatif stabil dalam kondisi dingin dan panas juga menjadi alasan tidak perlunya kalibrasi lebih lanjut terhadap sensor suhu ini.

#### 4.4. Pengujian Komunikasi Arduino Ethernet Shield dengan Web Server

Arduino ethernet shield pada sistem ini berfungsi sebagai media yang menghubungkan *control unit* dengan *web server* yaitu rephanuino.com. Untuk dapat melakukan fungsinya pada sistem, sebelumnya harus dilakukan pengujian terlebih dahulu terhadap kemampuan ethernet shield untuk menghubungkan *control unit* dengan server yang diinginkan melalui internet.



**Gambar 4.10.** Karakter yang Terbaca pada Laman Browser dengan URL:  
<http://rephanuino.com/BuildingAutomation/viewdbUI.php>.

```
COM8 (Arduino/Genuino Mega or Mega 2560)

connected
client disconnected.
Data from server captured in readString:

HTTP/1.1 200 OK
Server: openresty
Date: Tue, 17 Jul 2018 06:57:10 GMT
Content-Type: text/html; charset=UTF-8
X-Powered-By: PHP/7.0.30
Content-Length: 18
Connection: close

$a$b$c$d$e$f$g$h$i
End of readString
=====

connected
client disconnected.
Data from server captured in readString:

HTTP/1.1 200 OK
Server: openresty
Date: Tue, 17 Jul 2018 06:57:15 GMT
Content-Type: text/html; charset=UTF-8
X-Powered-By: PHP/7.0.30
Content-Length: 18
Connection: close

$a$b$c$d$e$f$g$h$i
End of readString
=====
```

Data string yang disimpan oleh arduino dari laman <http://rephanuino.com/BuildingAutomation/viewdbUI.php>

**Gambar 4.11.** Data String yang Terbaca oleh Arduino pada URL: <http://rephanuino.com/BuildingAutomation/viewdbUI.php>.

Pada gambar 4.10 dan 4.11 diperlihatkan objek visual sebagai indikator perbandingan antara karakter yang diperlihatkan pada

laman “<http://rephanuino.com/BuildingAutomation/viewdbUI.php>” dengan data berbentuk *string* yang disimpan pada arduino. Yang diinginkan oleh sistem adalah arduino dapat menyimpan karakter yang diperlihatkan pada laman *website* dalam bentuk *string* sehingga dapat dijadikan referensi pada operasi yang dilakukan oleh sistem.

Dari gambar 4.10 terlihat bahwa karakter yang diperlihatkan pada laman adalah “\$a\$b\$c\$d\$e\$f\$g\$h\$9”. Pada gambar 4.11 dapat dilihat bahwa serial monitor menampilkan data *string* yang disimpan dari “<http://rephanuino.com/BuildingAutomation/viewdbUI.php>”, dan data *string* yang ditampilkan sesuai dengan karakter yang ada pada gambar 4.10. Dari hasil yang didapatkan tersebut, dapat disimpulkan bahwa ethernet shield dapat menjalankan fungsinya dengan baik dan mampu menunjang fungsi koneksi terhadap *web server* yang telah dibuat pada sistem.

## **4.5. Pengujian Komunikasi Aplikasi Android dengan Web Server**

### **4.5.1. Percobaan Pengubahan Kondisi pada Aplikasi Android**

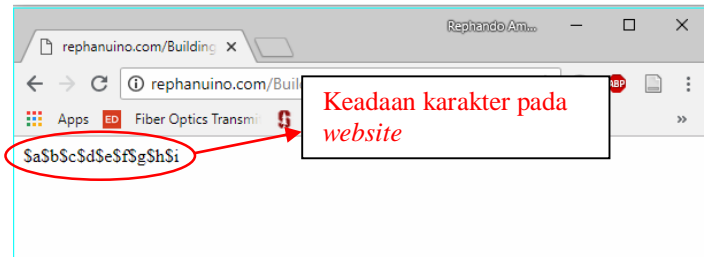
Aplikasi Android sebagai *command center* pada sistem ini berfungsi untuk memberikan perintah terhadap sistem melalui *web server*. Sebelum aplikasi ini digunakan dalam sistem, perlu dilakukan pengujian terlebih dahulu terhadap kemampuannya untuk berkomunikasi dengan *web server* melalui internet dan melakukan fungsi perintah.

Untuk itu dilakukan percobaan pengubahan keadaan tombol pada *command center* beberapa kali, kemudian dibandingkan dengan keadaan yang terlihat pada laman *website* yang ditunjukkan oleh URL “<http://rephanuino.com/BuildingAutomation/viewdbUI.php>”. Di mana laman tersebut menampilkan karakter yang tersimpan pada tabel kondisi yang tersimpan di *database web server*. Bila sistem bekerja dengan seharusnya, nilai yang ditampilkan pada laman tersebut seharusnya berubah sesuai dengan perintah yang diberikan oleh *command center* (aplikasi android) lewat penekanan tombol yang ada.





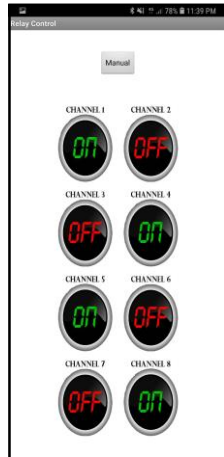
(a)



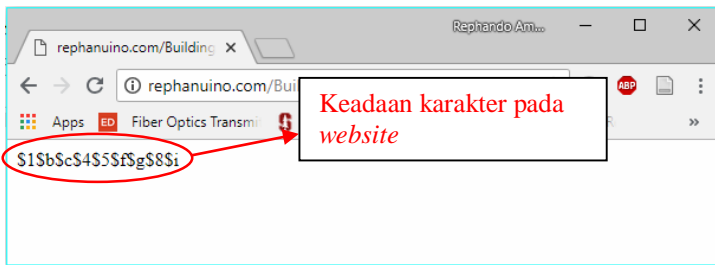
(b)

**Gambar 4.12.** (a) Keadaan Tombol pada Aplikasi; (b) Keadaan Karakter pada *Web Server* pada Percobaan Pertama

Pada gambar 4.12 diperlihatkan hubungan antara keadaan tombol pada aplikasi dengan keadaan karakter pada *web server*. Pada gambar 4.12 (a) terlihat bahwa tombol mode pada aplikasi menunjukkan keadaan “manual” dan semua tombol kontrol *channel* dalam keadaan “OFF”. Pada gambar 4.12 (b) terlihat bahwa keadaan pada *web server* menunjukkan deret karakter sebagai berikut: “\$a\$b\$c\$d\$e\$f\$g\$h\$i”.



(a)



(b)

**Gambar 4.13.** (a) Keadaan Tombol pada Aplikasi; (b) Keadaan Karakter pada *Web Server* pada Percobaan kedua.

Pada gambar 4.13 (a) terlihat bahwa tombol mode pada aplikasi menunjukkan keadaan “manual”. Tombol kontrol *channel* 1, 4, 5, dan 8 dalam keadaan “ON”, sedangkan tombol kontrol *channel* 2, 3, 6, dan 7 dalam keadaan “OFF”. Pada gambar 4.13 (b) terlihat bahwa keadaan pada *web server* menunjukkan deret karakter sebagai berikut: “\$1\$b\$c\$4\$5\$f\$g\$8\$i”, dimana ada karakter yang berubah dibandingkan dengan deret karakter yang terdapat pada gambar 4.12. Angka 1, 4, 5, 8 yang terdapat pada deret karakter di *web server* merepresentasikan tombol pada keadaan “ON” pada aplikasi android.

#### 4.5.2. Analisis Terhadap Komunikasi Aplikasi Android dengan Web Server

Dari hasil percobaan yang telah dilakukan dengan mengubah-ubah keadaan pada tombol aplikasi android, dapat disimpulkan bahwa aplikasi android dapat bekerja dengan baik karena menghasilkan hasil yang diinginkan saat keadaan pada tombol diubah-ubah.

### 4.6. Pengujian Mode Operasi

Sistem ini memiliki dua mode operasi seperti yang telah dijelaskan sebelumnya. Pada bagian ini diperlihatkan hasil dari pengujian mode operasi pada sistem, mode operasi manual dan mode operasi *auto*.

#### 4.6.1. Mode Operasi Manual

Mode operasi manual pada sistem berjalan dengan baik, dan respon terhadap perintah yang diberikan relatif cepat. Pada operasi manual ini, seperti yang telah dijelaskan sebelumnya, *control unit* langsung bekerja berdasarkan perintah dari *command center* yang dikendalikan oleh *user* secara langsung berupa aplikasi android.



**Gambar 4.14.** Pengujian Operasi Mode Manual Pertama.



**Gambar 4.15.** Pengujian Operasi Mode Manual Kedua.



**Gambar 4.16.** Pengujian Operasi Mode Manual Ketiga.

Pada gambar 4.14, gambar 4.15, dan gambar 4.16 diperlihatkan metode percobaan operasi manual yang dilakukan terhadap sistem. Percobaan dilakukan dengan menggunakan 5 buah lampu LED 3 Watt yang terhubung dengan pada *channel* 1, 2, 3, 4, dan 5 pada *control unit* sistem. Pada percobaan ini tombol mode pada aplikasi android di-*set* pada keadaan “Manual”. Kemudian layar LCD pada

*control unit* menampilkan “Manual Mode” yang mengindikasikan bahwa sistem sekarang bekerja dalam mode manual.

Pada gambar 4.14, keadaan tombol kendali operasi manual untuk *channel 1* hingga *channel 5* pada *command center* dalam keadaan: *channel 1* dan *channel 5* “ON”, lalu *channel 2*, *channel 3*, dan *channel 4* dalam keadaan “OFF”. Dan di sisi keluaran sistem, lampu pada *channel 1* dan *channel 5* dalam keadaan menyala dan *channel* lainnya mati.

Pada gambar 4.15 keadaan tombol kendali operasi manual dalam keadaan: *channel 1*, *channel 3*, dan *channel 5* “ON”, dan *channel* yang lainnya “OFF”. Pada realisasinya terhadap 5 buah lampu LED yang dihubungkan terlihat bahwa keadaan lampu pada *channel 1*, *channel 3*, dan *channel 5* dalam keadaan menyala dan yang lainnya dalam keadaan mati.

Percobaan terakhir yang dapat dilihat pada gambar 4.16 memperlihatkan tombol pada aplikasi yang dalam keadaan “ON” semua untuk *channel 1* hingga *channel 5* (yang digunakan dalam percobaan). Dan hasil realisasi pada keadaan lampu LED yang dihubungkan pada *channel 1* hingga *channel 5* pada *control unit* adalah dalam keadaan hidup semua.

Dari 3 percobaan yang dilakukan dalam keadaan yang berbeda-beda, dapat terlihat bahwa sistem melakukan fungsi operasi manual sesuai dengan yang seharusnya terjadi. Hanya saja pada pengaplikasiannya kontrol manual ini dibatasi oleh kebutuhan akan adanya koneksi internet, dan fungsinya akan tidak maksimal jika koneksi internet yang digunakan oleh sistem kurang lancar atau terputus-putus.

#### **4.6.2. Mode Operasi Auto**

Mode operasi *auto* pada sistem ini berjalan dengan baik juga, sama dengan kontrol manual yang dijelaskan pada bagian sebelumnya. Mode *auto* pada sistem ini bekerja dengan menghubungkan aktuator (*relay*) dengan input dari sensor cahaya dan juga sensor suhu yang dihubungkan pada *control unit* melalui rangkaian *sensor dock*.

Pertama-tama dilakukan percobaan pengaruh sensor cahaya pada aktivitas *control unit* dalam mode *auto*.



**Gambar 4.17.** Keadaan Saat Cahaya di Bawah *Bottom Limit*.



**Gambar 4.18.** Keadaan Saat Cahaya di Antara *Bottom Limit* dan *Upper Limit*(1).



**Gambar 4.19.** Keadaan Saat Cahaya di Atas *Upper Limit*.



**Gambar 4.20.** Keadaan Saat Cahaya di Antara *Bottom Limit* dan *Upper Limit*(2).

Pada gambar 4.17 hingga gambar 4.20 diperlihatkan proses percobaan mode *auto* dengan pengaruh sensor cahaya secara

bertahap. Pertama keadaan ruangan diberikan intensitas cahaya di bawah *lower limit* yang telah ditentukan melalui fungsi “*settings*” (gambar 4.17), lantas lampu pada *channel* yang bersangkutan menyala. Kemudian intensitas cahaya dinaikkan, tetapi tidak mencapai *upper limit* yang telah ditentukan (gambar 4.18), dan lampu masih dalam keadaan menyala. Setelah itu intensitas cahaya ruangan dinaikkan lagi hingga di atas *upper limit* yang ditentukan (gambar 4.19), dan lampu dalam keadaan menyala. Terakhir, intensitas cahaya diturunkan kembali hingga di antara *upper limit* dan *lower limit* (gambar 4.20), lampu pun tetap dalam keadaan mati.

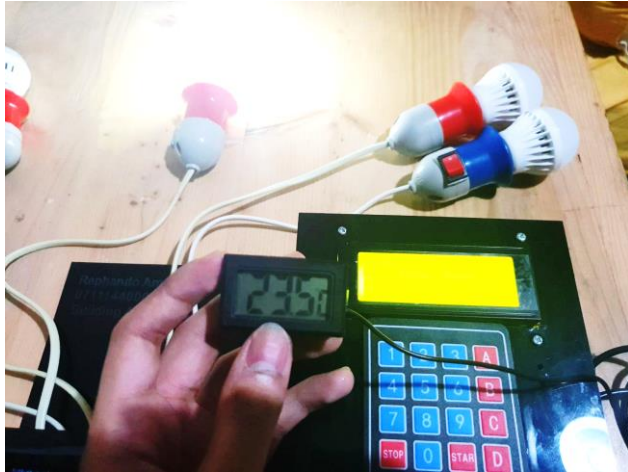
Hal ini sesuai dengan prinsip kerja yang ditentukan, lampu akan menyala hanya saat intensitas cahaya yang diterima sensor memiliki nilai di bawah *lower limit*, dan akan menyala hanya saat intensitas cahaya yang diterima sensor melebihi *upper limit*.

Kemudian dilakukan pula percobaan mode *auto* berdasarkan *input* dan sensor suhu.

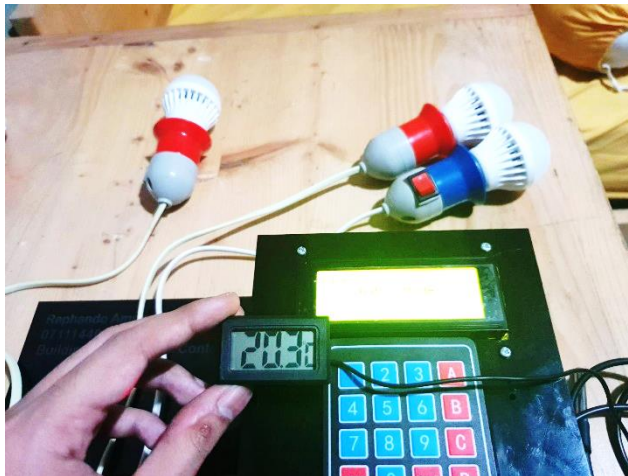


**Gambar 4.21.** Keadaan Saat Suhu Ruangan di Atas *Upper Limit*.





**Gambar 4.22.** Keadaan Saat Suhu Ruangan di Antara *Bottom Limit* dan *Upper Limit*.



**Gambar 4.23.** Keadaan Saat Suhu Ruangan di Bawah *Bottom Limit*.

Gambar 4.21 hingga gambar 4.23 menunjukkan tahapan dari pengujian mode *auto* berdasarkan *input* dari sensor suhu. Pada

percobaan ini *bottom limit* suhu ditentukan pada angka 22°C, dan *upper limit* suhu ditentukan pada angka 30°C. Lampu dijadikan objek yang dikontrol karena mudah untuk diobservasi dan perubahannya terlihat. Pada gambar 4.21 ruangan ada pada keadaan suhu yang di atas *upper limit* yang ditentukan, kemudian lampu menyala. Setelah itu pada gambar 4.22 ruangan berada pada keadaan nilai suhu di antara *upper limit* dan *lower limit*, lampu masih dalam keadaan menyala saat ini. Kemudian suhu diturunkan hingga di bawah dari *lower limit* yang ditentukan pada gambar 4.23, dan lampu pun mati.

Pola kerja yang terjadi pada percobaan ini sesuai dengan prinsip yang ditentukan, bahwa alat yang terintegrasi dengan sensor suhu akan menyala saat suhu riil berada di atas nilai *upper limit* yang ditentukan, dan mati saat suhu riil berada di bawah nilai *lower limit* yang ditentukan.

#### **4.7. Analisis Keseluruhan Sistem**

Secara keseluruhan, modul sistem otomasi bangunan yang telah direalisasikan melalui penelitian ini dapat menjalankan fungsinya dengan baik sesuai dengan tujuan perancangan. Dari percobaan yang telah dilakukan, didapatkan adanya *delay* yang dibutuhkan oleh arduino untuk mengambil data dari server dan mengolahnya, sebelum mengambil data yang baru.

**Tabel 4.7.** Data Pengujian *Delay* Mode Operasi Manual

No	Interval Pengambilan Data(s)
1	0,702
2	0,79
3	3,912
4	0,685
5	0,646
6	0,63
7	0,667
8	0,606
9	0,758
10	0,813
11	0,527
12	0,964
13	0,698
14	0,729
15	0,605
16	0,73
17	0,723
18	0,699
<b>Rata-rata</b>	<b>0,8824</b>

**Tabel 4.8.** Data Pengujian *Delay Mode Operasi Auto*

No	Interval Pengambilan Data(s)
1	2,132
2	2,58
3	2,799
4	2,59
5	2,8
6	2,524
7	2,795
8	2,636
9	2,598
10	2,674
11	2,666
12	5,865
13	2,508
14	2,833
15	5,824
16	2,578
17	2,544
18	2,637
<b>Rata-rata</b>	<b>2,97683</b>

*Delay* yang tercatat setelah melakukan percobaan pada mode operasi manual menghasilkan nilai rata-rata sebesar 0,88s dan pada mode operasi *auto* menghasilkan nilai rata-rata sebesar 2,97s. Pengambilan data *delay* dilakukan dengan cara mengambil *video* saat sistem sedang aktif mengambil data dari *server* dan mengamati interval data yang masuk dengan ketelitian 1ms. Pada mode *auto* memiliki *delay* yang lebih besar dikarenakan perlunya waktu tambahan saat mengambil nilai sensor suhu DS18B20.

Dari percobaan yang dilakukan, *control unit* pada sistem ini membutuhkan daya sebesar 5,2 Watt untuk dalam keadaan *standby*.

Hal ini kemudian ditabrakkan dengan pernyataan dari kementerian ESDM pada tanggal 27 Oktober 2016 tentang “vampir listrik” atau “*standby power*” yang sering diabaikan oleh masyarakat Indonesia. Tim komunikasi kementerian ESDM menyebutkan bahwa rata-rata masyarakat membiarkan daya sebesar 83 Watt terbuang sebagai “vampir listrik”[5]. Bila kelistrikan dalam suatu rumah dilengkapi dengan modul sistem otomasi bangunan ini, masyarakat dapat menghemat energi listrik karena “vampir listrik” tereliminasi dan daya yang dibutuhkan sebagai kompensasinya adalah sebesar 5,2 Watt.



**Gambar 4.24.** Daya Listrik yang Dibutuhkan Oleh *Control Unit* Sistem.

.....*Halaman ini sengaja dikosongkan*.....

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Dari percobaan dan hasil yang telah didapat pada proses pengerjaan penelitian sistem ini, dapat diambil beberapa poin kesimpulan:

- a. Modul sistem otomasi bangunan ini merupakan sistem yang sangat relevan untuk diaplikasikan pada masyarakat di era ini, khususnya masyarakat perkotaan yang memiliki tingkat kesibukan yang tinggi.
- b. Sistem *building automation* yang terbagi ke dalam 3 bagian yaitu *control unit*, *command center*, dan *web server* ini berfungsi sesuai dengan tujuan perancangannya.
- c. Sistem *building automation* ini merupakan sistem yang *user-friendly* dan dapat digunakan dan dioperasikan oleh masyarakat awam.
- d. Sistem *building automation* ini dapat melakukan fungsi penghematan energi dengan mengeliminasi *standby power* atau vampir listrik.

#### **5.2. Saran**

Setelah mendapatkan hasil dari proses pelaksanaan tugas akhir ini, penulis memberikan beberapa saran dalam rangka menunjang pengembangan sistem di kemudian hari:

- a. Parameter kondisi lingkungan yang memengaruhi kerja sistem pada mode *auto* dapat ditambah selain cahaya dan suhu, agar sifat serba guna dari sistem semakin terlihat dan berpengaruh.
- b. Aktuasi yang dilakukan sistem terhadap perangkat-perangkat elektronik eksternal dapat dikembangkan di luar fungsi *on/off* agar pengembangan sistem cakupannya lebih luas.

.....*Halaman ini sengaja dikosongkan*.....



## DAFTAR PUSTAKA

- [1] “Pemerintah Penuhi Target Rasio Elektrifikasi, Jonan: No One Left Behind,” *ESDM*. [Online]. Available: <https://www.esdm.go.id/id/media-center/arsip-berita/pemerintah-penuhi-target-rasio-elektrifikasi-jonan-no-one-left-behind>. [Accessed: 24-Apr-2018].
- [2] M. Ersue, D. Romascanu, J. Schoenwaelder, and A. Sehgal, “Management of Networks with Constrained Devices: Use Cases,” Jul. 2014.
- [3] D. Evans, “How the Next Evolution of the Internet Is Changing Everything,” p. 11, 2011.
- [4] “Indonesia is the 3rd-Largest Smartphone Market in the Asia Pacific | Indonesia Investments.” [Online]. Available: <https://www.indonesia-investments.com/news/todays-headlines/indonesia-is-the-3rd-largest-smartphone-market-in-the-asia-pacific/item6777?> [Accessed: 03-Jun-2018].
- [5] “Listrik Baik untuk Indonesia Mandiri Energi,” *ESDM*. [Online]. Available: <https://www.esdm.go.id/en/media-center/news-archives/listrik-baik-untuk-indonesia-mandiri-energi>. [Accessed: 03-Jun-2018].
- [6] “Building Automation system,” *TechiExpert*, 07-Dec-2016. .
- [7] A. Ozadowicz and J. Grela, “Impact of building automation control systems on energy efficiency #x2014; University building case study,” in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–8.
- [8] “Understanding Building Automation and Control Systems -KMC Controls,” 19-May-2013. [Online]. Available: [https://web.archive.org/web/20130519124213/http://www.kmccontrols.com/products/Understanding\\_Building\\_Automation\\_and\\_Control\\_Systems.aspx](https://web.archive.org/web/20130519124213/http://www.kmccontrols.com/products/Understanding_Building_Automation_and_Control_Systems.aspx). [Accessed: 11-May-2018].
- [9] M. Drăgoicea, L. Bucur, and M. Pătrașcu, “A Service Oriented Simulation Architecture for Intelligent Building Management,” in *Exploring Services Science*, 2013, pp. 14–28.
- [10] “Control Solutions | The Ultimate Guide to Building Automation,” *Control Solutions*. [Online]. Available: <http://controlyourbuilding.com/index.php/blog/entry/the-ultimate->

- guide-to-building-automation. [Accessed: 12-May-2018].
- [11] R. Zafalon, "Smart System Design: Industrial Challenges and Perspectives," in *2013 IEEE 14th International Conference on Mobile Data Management*, 2013, vol. 1, pp. 3–3.
  - [12] D. R. M. Makwana and D. R. M. Makwana, "Major Applications and Issues in Internet of Things," *AleHive.com*, 01-Oct-2016. .
  - [13] "Arduino Mega 2560 Rev3." [Online]. Available: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>. [Accessed: 12-May-2018].
  - [14] "[SINTRON] Arduino MEGA 2560 R3 + USB cable + reference PDF files for starter-in Replacement Parts from Consumer Electronics on Aliexpress.com | Alibaba Group," *aliexpress.com*. [Online]. Available: [//www.aliexpress.com/item/SINTRON-Arduino-MEGA-2560-R3-USB-cable-reference-PDF-files-for-starter/878813783.html?src=ibdm\\_d03p0558e02r02](https://www.aliexpress.com/item/SINTRON-Arduino-MEGA-2560-R3-USB-cable-reference-PDF-files-for-starter/878813783.html?src=ibdm_d03p0558e02r02). [Accessed: 14-May-2018].
  - [15] M. Kusriyanto and B. D. Putra, "Smart home using local area network (LAN) based arduino mega 2560," in *2016 2nd International Conference on Wireless and Telematics (ICWT)*, 2016, pp. 127–131.
  - [16] "Arduino - Ethernet." [Online]. Available: <https://www.arduino.cc/en/Reference/Ethernet>. [Accessed: 18-May-2018].
  - [17] "Arduino - ArduinoEthernetShieldV1." [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>. [Accessed: 19-May-2018].
  - [18] "Arduino - Environment." [Online]. Available: <https://www.arduino.cc/en/Guide/Environment>. [Accessed: 30-Jan-2018].
  - [19] "DS18B20 Water-Proof Temperature Sensor Probe," *PotentialLabs*. [Online]. Available: <https://potentiallabs.com/cart/buy-ds18b20-waterproof-online-hyderabad-india>. [Accessed: 02-Jun-2018].
  - [20] R. G. Useinov, O. V. Meshurov, M. G. Drosdetsky, and G. I. Zebrev, "Simulation of radiation-induced supply leakage currents in modern digital CMOS thermometer DS18B20," 2018, pp. 1–3.
  - [21] "DS18B20-Programmable Resolution 1-Wire Digital Thermometer." [Online]. Available: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.

- [Accessed: 25-May-2018].
- [22] I. Amri, E. D. Atmajati, R. A. Salam, E. Yuliza, M. M. Munir, and Khairurrijal, "Potentiometer a simple light dependent resistor-based digital," in *2016 International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM)*, 2016, pp. 24–27.
  - [23] "LDR - Light Dependent Resistor - Large," *PotentialLabs*. [Online]. Available: <https://potentiallabs.com/cart/ldr-big>. [Accessed: 01-Jun-2018].
  - [24] M. Jain, N. Kaushik, and K. Jayavel, "Building automation and energy control using IoT - Smart campus," in *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, 2017, pp. 353–359.
  - [25] "RELAY (12VDC 10A)," *Lampa Tronics*. .
  - [26] "Bembyho blog | Controlling 5V relay with Raspberry." [Online]. Available: <http://www.bambusekd.cz/dev/raspberry-control-5V-relay>. [Accessed: 18-May-2018].
  - [27] "About Us | Explore MIT App Inventor." [Online]. Available: <http://appinventor.mit.edu/explore/about-us.html>. [Accessed: 20-May-2018].
  - [28] "Android, Where's My Car? for App Inventor 2 | Explore MIT App Inventor." [Online]. Available: <http://appinventor.mit.edu/explore/ai2/android-wheres-my-car.html>. [Accessed: 20-May-2018].
  - [29] "Ruan paper CTE.pdf," *Google Docs*. [Online]. Available: [https://drive.google.com/file/d/0B8AUNoeCnCCARUMtS0NqN3R5d1E/view?usp=sharing&usp=embed\\_facebook](https://drive.google.com/file/d/0B8AUNoeCnCCARUMtS0NqN3R5d1E/view?usp=sharing&usp=embed_facebook). [Accessed: 20-May-2018].
  - [30] "Autodesk EAGLE Now Available | EAGLE | Blog," *Eagle Blog*, 16-Jan-2017. [Online]. Available: <https://www.autodesk.com/products/eagle/blog/autodesk-eagle-has-landed-the-easy-pcb-design-software-now-under-the-autodesk-wing/>. [Accessed: 20-May-2018].
  - [31] "Definition of LCD." [Online]. Available: <https://www.merriam-webster.com/dictionary/LCD>. [Accessed: 15-Jun-2018].
  - [32] A. Industries, "Standard LCD 20x4 + extras." [Online]. Available: <https://www.adafruit.com/product/198>. [Accessed: 22-May-2018].
  - [33] E. Latorres, "Arduino I2C-LCD Display Project with Malpartida Library," *Enrique Latorres*. .

.....*Halaman ini sengaja dikosongkan*.....

**LAMPIRAN A**

**Tanggapan Responden Terhadap Pertanyaan**  
**“Apakah harapan anda terhadap perkembangan teknologi otomasi**  
**bangunan?”**  
**Pada Kuesioner**

No	TANGGAPAN
1	Harapan saya perlu adanya update berkala pada pengembangan aplikasi dalam jangka waktu pendek agar dapat memenuhi keinginan pengguna
2	Otomasi sangat diperlukan kedepannya, namun harus diberi batasan untuk pengembangannya. Melihat semakin banyak orang yang malas untuk bergerak sehingga angka obesitas dapat meningkat. Ketika teknologi otomasi dikembangkan, maintenancenya juga harus diperhatikan. Karena salah satu kendala terbesar dari teknologi otomasi pasti masalah maintenance, melihat sering tidak adanya backup mekanik ketika sistem otomasi tersebut tidak dapat digunakan.
3	Mudah dioperasikan & bisa membantu kegiatan-kegiatan keseharian kita
4	Perlu dikembangkan
5	Otomasi membantu kontrol aktivitas rutin lebih intens
6	Belum merasa perlu
7	Meningkatkan keselamatan, kemudahan pengguna
8	Teknologi banyak menggunakan sensor gerak
9	Bisa dengan mudah memantau keadaan rumah dari mana saja. Seperti yg sudah ada sekarang (appliances rmh bs diakses dr jauh, melalui gadget) Hanya saja lebih disederhanakan. Melihat penggunaan listrik terbesar bersumber dari appliances yg mana, air panas kmr mandi yg bisa langsung panas, tanpa membuang air, dan dapat mengatur akses masuk ke rumah, bila ada anggota keluarga yang tidak bawa kunci
10	Bisa mempermudah masyarakat

11	Perlu dikembangkan fitur baru utk fungsi otomatisasi dan desain tampilan yang mudah.
12	Perlu dikembangkan. Namun saya berharap scope nya dibatasi tidak dalam segala hal.
13	Ingin ada alat otomatis saat ada bahaya
14	Lebih baik
15	Belum, manual sangat membantu mobilisasi gerak tubuh - automation akan merubah pola hidup generasi depan , kecuali yg sewaktu waktu.
16	Saya sangat berharap teknologi otomasi bangunan dapat dikembangkan lebih lanjut untuk memudahkan kegiatan sehari-hari di rumah.
17	Remember if I forget to stop the electricity and I can reduce to used it
18	Mendukung kualitas hidup bagi lansia yang hidup sendiri di rumah. Khususnya area kamar mandi, dapur dan tangga/level lantai.
19	Perlu dikembangkan teknologi otomasi dalam bangunan untuk memudahkan mengontrol pemakaian listrik dengan switch on/off yg tersentralisasi tapi fleksibel bisa diakses dari mana saja, apalagi jika terdapat banyak ruang dalam rumah atau pada high building.
20	Smart technology
21	Dapat diaplikasikan dengan biaya seminimum mungkin dan terjangkau terutama utk kebutuhan rumah tangga serta dg interface yg tidak terlalu rumit bagi ibu rmh tangga yg super sibuk
22	Tombol yang lebih ergonomis dan modern serta sistem meteran listrik yang terintegrasi dengan smartphone dan aplikasinya
23	Perlu dikembangkannya perintah suara dalam penggunaan peralatan elektronik agar memudahkan pengguna.
24	Semakin mempermudah
25	Pengaplikasiannya agar dapat diterapkan di hampir setiap alat elektronik

26	Adanya pengembangan otomasi pada tiap peralatan agar usernya dapat lebih nyaman dan aman.
27	Semua diserahkan untuk kemuliaan Tuhan Yesus Kristus.
28	Interface menarik dan mudah dipahami
29	Perlu dikembangkan
30	Saya berharap bisa mengendalikan peralatan dirumah menggunakan pikiran
31	Saya rasa sudah ada ya teknologi lampu atau AC yang dapat dikontrol melalui aplikasi, hanya saja mungkin harga yg cukup mahal membuatnya jarang diterapkan. Tapi sebenarnya teknologi ini sangat dibutuhkan. Mari kita tengok jurusan kita sendiri, setelah kelas selesai, lampu dan AC tetap dibiarkan menyala meskipun tidak ada manusia di dalam ruangan, hal ini tentu merupakan tindakan pemborosan energi. Kenapa tidak dimatikan? Mungkin lupa atau kurang peduli, di sini lah peran otomasi bangunan sangat dibutuhkan.
32	Fitur Schedule atau Otomatis menggunakan sensor
33	Peralatan elektronik dapat digunakan semudah menulis tweet di Twitter
34	Memiliki fitur monitoring beban, fitur load forecasting yang diintegrasikan dengan tagihan PLN, dan fitur smart switch.
35	Smart Home bisa luas diaplikasikan di Indonesia
36	Semoga budaya2 kita tidak luntur seiring dengan perkembangan zaman
37	Sangat perlu ditingkatkan dan dikembangkan menilik perkembangan zaman akan energi listrik yang sangat berkembang disekor manapun
38	IoT dan otomasi diharapkan dapat diterapkan secara lebih luas pada kehidupan rumah tangga untuk mengurangi kelalaian yang disebabkan oleh manusia sehingga efisiensi dan efektifitas dalam pemakaian peralatan elekteonik
39	Teknologi otomasi bangunan dapat dikendalikan melalui sensor gerak, wifi. Dan harga juga terjangkau
40	Rumah bisa berbicara ke penghuni tentang kondisi terkini si penghuni misalkan "anda harus istirahat, matikan lampu"

41	Adanya apps untuk memonitoring peralatan elektoonik di rumah. Salah satunya menyalakan dan mematikan lampu. Mangaatss
42	menggunakan sistem IOT, keamanan ditingkatkan
43	Harapannya teknologi otomasi bangunan dapat memberikan kemudahan dalam mengatur penggunaan energi sehingga energi listrik bisa dimanfaatkan lebih efisien.
44	Perlu dikembangkan agar dapat mempermudah kehidupan
45	Semoga semakin mudah
46	Semoga teknologi otomasi bangunan dapat dikembangkan kedepannya, agar supaya peralatan elektronik dapat diaplikasikan secara otomatis / melalui smartphone karena hal tersebut dapat mempermudah pekerjaan manusia
47	Harapannya sinyal dari jaringan wifi tdk berpengaruh negatif terhadap pertumbuhan/perkembangan otak
48	Dapat memantau dari jarak jauh
49	Bangunan dapat menyerap gas CO2
50	Semakin otomatis, dan cerdas Pemutus ketika harga listrik sudah mencapai batas mahal
51	Bisa dipertimbangkan juga untuk efiesiesi biaya pengeluaran listrik. Hemat energi jadinya pasti hemat biaya..
52	Dapat mempermudah kontrol peralatan elektronik dirumah tanpa resah dan mudah.
53	Bisa mengoperasikan alat elektronik yang ada di rumah dari jarak jauh, misalnya menyalakan ac dalam waktu 5 menit sebelum penghuni sampai di rumah sehingga saat ia sampai ruangan tidak terasa panas. atau bisa juga setiap alat elektronik dilengkapi dengan sensor supaya saat penghuni meninggalkan suatu ruangan, lampu akan mati secara otomatis guna menghemat penggunaan listrik
54	Aspek sekuritas akan menjadi sangat penting apabila nantinya akan menggunakan internet untuk pemantauan dan pengendalian alat-alat elektronik. Dan penambahan fitur backup control unit maupun manual override apabila sistem otomasi ini mengalami masalah.



55	Otomasi melalui sinyal otak tanpa media H dan lainnya
56	Fitur bersihin rumah otomatis, sama fitur keamanan
57	Mudah dioperasikan,dapat melihat kondisi bangunan(peralatan) walau sedang jauh dan dapat mengendalikannya
58	Pengen punya lampu yg bisa nyala/mati tergantung ada orngnya atau ngga
59	User friendly
60	Selalu berkembang dan memanjakan penghuni rumah.
61	Perlu dikembangkan mengenai teknologi otomasi bangunan mengingat dengan berkembangnya teknologi saat ini diharapkan segala sesuatunya dapat dilakukan dengan mudah, cepat, dan efisien melalui internet
62	Bisa memudahkan urusan sehari-harinya
63	Antisipasi saat terjadi pemadaman listrik
64	Memberikan kemudahan dalam pengoperasian peralatan
65	peralatan elektronik yang bisa diatur melalui smartphone
66	Adanya informasi secara langsung tentang penggunaan peralatan listrik dan bisa terjangkau oleh semua kalangan.
67	Agar tidak menjadi beban pikiran pada saat kita diluar rumah
68	Perlu dikembangkan khususnya dari segi lifetime serta keamanan alatnya
69	Lebih berkembang
70	Adanya teknologi yg "ramah kantong" untuk otomasi bangunan
71	Rumah dapat mengidentifikasi subyek yang berada di dalam rumah, serta dapat mengklasifikasikan aksi (action) yang dilakukan oleh subyek sehingga rumah dapat memperkirakan tingkat aktifitas subyek dan dapat mengatur kondisi lampu di dalam rumah sesuai dengan tingkat aktifitas subyek. Rumah juga diharapkan memiliki sistem keamanan yang bersumber dari identifikasi subyek yang berada di dalam rumah. Otomasi rumah terutama lampu tidak hanya diatur oleh keberadaan subyek, tetapi juga dari tingkat aktifitasnya.
72	User friendly

73	Dapat diaplikasikan langsung / menggantikan perangkat sebelumnya tanpa mengubah instalasi kabel
74	Semakin mudah didapatkan terutama masyarakat yang masih awam terhadap teknologi
75	Mudah dipasang sendiri dan UI mudah dimengerti. Keamanan jaringan terjamin.

## LAMPIRAN B

**Tabel Penggunaan Pin Pada Arduino MEGA 2560**

No.	Pin Arduino	Koneksi Komponen	Keterangan
1	A0	Sensor Cahaya 1	Membaca nilai ADC dari sensor cahaya
2	A1	Sensor Cahaya 2	
3	A2	Sensor Cahaya 3	
4	A3	Sensor Cahaya 4	
5	A4	Sensor Cahaya 5	
6	D42	Sensor Suhu 1	Membaca nilai dari sensor suhu
7	D44	Sensor Suhu 2	
8	D46	Sensor Suhu 3	
9	D22	<i>Relay Channel 1</i>	Mengaktifkan dan menonaktifkan <i>relay</i>
10	D24	<i>Relay Channel 2</i>	
11	D26	<i>Relay Channel 3</i>	
12	D28	<i>Relay Channel 4</i>	
13	D30	<i>Relay Channel 5</i>	
14	D32	<i>Relay Channel 6</i>	
15	D34	<i>Relay Channel 7</i>	
16	D36	<i>Relay Channel 8</i>	
17	D31	<i>Keypad Row 1</i>	Melihat keadaan tombol pada <i>membrane keypad 4x4</i>
18	D33	<i>Keypad Row 2</i>	
19	D35	<i>Keypad Row 3</i>	
20	D37	<i>Keypad Row 4</i>	
21	D39	<i>Keypad Column 1</i>	
22	D41	<i>Keypad Column 2</i>	
23	D43	<i>Keypad Column 3</i>	
24	D45	<i>Keypad Column 4</i>	
25	SCL	I <sup>2</sup> C LCD	Untuk memberikan perintah terhadap LCD
26	SDA		
27	D50	MISO	Komunikasi dengan <i>ethernet shield</i>
28	D51	MOSI	
29	D52	SCK	
30	D10	Select W5100	Memilih processor untuk aktif
31	D4	Select SD Card	Memiliki SD card untuk aktif

.....*Halaman ini sengaja dikosongkan*.....

## LAMPIRAN C

### Program Arduino pada *Control Unit Sistem Building Automation*

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SPI.h>
#include <Ethernet.h>
#include <LiquidCrystal_I2C.h> //include LCD library (standard library)
#include <Keypad.h> //include keypad library - first you must install library (library link in
the video description)

//For Keypad-----
int pozisyon = 0; //keypad position
const byte rows = 4; //number of the keypad's rows and columns
const byte cols = 4;

char keyMap [rows] [cols] = { //define the symbols on the buttons of the keypad

    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

char whichKey;

byte rowPins [rows] = { 31, 33, 35, 37 }; //pins of the keypad
byte colPins [cols] = { 39, 41, 43, 45 };
Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, rows, cols);
//For Keypad-----

//For Internet communication-----
String readString;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //physical mac address

char serverName[] = "www.rephanuino.com"; // rephanuino web page server
String arrayon[] = { "$1", "$2", "$3", "$4", "$5", "$6", "$7", "$8", "$9" };
String arrayoff[] = { "$a", "$b", "$c", "$d", "$e", "$f", "$g", "$h", "$i" };
EthernetClient client;
//For Internet communication-----
//int AutoPin = digitalRead(40);
//int ManualPin = digitalRead(38);

//variabel penyimpan nilai relay n sensor-----
int r[8];
int s[5];
```

```

int t[5];
int x;
int y;
int LightBottomLimit[5];
int LightUpperLimit[5];
float TempBottomLimit[5];
float TempUpperLimit[5];
String strTemp = "";

// int LightSensor1 = A0;
// int LightSensor2 = A1;
// int LightSensor3 = A2;
// int LightSensor4 = A3;
// int LightSensor5 = A4;
int LightValue[6];
//
// int TempSensor1 = A8;
// int TempSensor2 = A9;
// int TempSensor3 = A10;
float TempValue[4];
//variabel penyimpan nilai relay n sensor-----

//variabel channel relay-----
int RelayChannel[] = {99, 22, 24, 26, 28, 30, 32, 34, 36};
//variabel channel relay-----

// Data wire is plugged into port 2 on the Arduino
#define TempSensor1 42
#define TempSensor2 44
#define TempSensor3 46

// Setup a oneWire instance to communicate with any OneWire devices (not just
Maxim/Dallas temperature ICs)
OneWire oneWire1(TempSensor1);
OneWire oneWire2(TempSensor2);
OneWire oneWire3(TempSensor3);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors1(&oneWire1);
DallasTemperature sensors2(&oneWire2);
DallasTemperature sensors3(&oneWire3);

LiquidCrystal_I2C lcd(0x3F, 20, 4);

void setup()
{
  Serial.begin(9600);

  lcd.init();

```

```

lcd.init();
lcd.backlight();

pinMode(RelayChannel[0], OUTPUT);
pinMode(RelayChannel[1], OUTPUT);
pinMode(RelayChannel[2], OUTPUT);
pinMode(RelayChannel[3], OUTPUT);
pinMode(RelayChannel[4], OUTPUT);
pinMode(RelayChannel[5], OUTPUT);
pinMode(RelayChannel[6], OUTPUT);
pinMode(RelayChannel[7], OUTPUT);
pinMode(RelayChannel[8], OUTPUT);

sensors1.begin(); //initialize the three DS18B20
sensors2.begin();
sensors3.begin();
Ethernet.begin(mac);

Serial.println("client readString test 11/04/13"); // so I can keep track of what is loaded
}

void loop()
{
  lcd.setCursor(0, 0);
  lcd.print("      ");

  InternetConnection();

  if (digitalRead(38))
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PING");
    lcd.setCursor(5, 1);
    lcd.print("Auto   Mode");
    delay(100);
    AutoMode();
  }
  else
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PING");
    lcd.setCursor(4, 1);
    lcd.print("Manual  Mode");
    delay(100);
    ManualMode();
  }
}

```

```

whichKey = myKeypad.getKey();
if (whichKey == '#')
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sensor");
    lcd.setCursor(0, 1);
    lcd.print("Settings");
    delay(2000);
    lcd.clear();
    settings();
}
else if (whichKey == 'A')
{
    MonitorCahaya();
}
else if (whichKey == 'B')
{
    MonitorSuhu();
}
whichKey = myKeypad.getKey();
}

//=====================================================
void InternetConnection() //client function to send/receive GET request data.
{
    if (client.connect(serverName, 80)) { //starts client connection, checks for connection
        Serial.println("connected");
        client.println("GET /BuildingAutomation/viewdbUI.php HTTP/1.1"); //download text
        client.println("Host: www.rephanuino.com");
        client.println("Connection: close"); //close 1.1 persistent connection
        client.println(); //end of get request
    }
    else {
        Serial.println("connection failed"); //error message if no client connect
        Serial.println();
    }

    while (client.connected() && !client.available()) delay(1); //waits for data
    while (client.connected() || client.available()) { //connected or data available
        char c = client.read(); //gets byte from ethernet buffer
        readString += c; //places captured byte in readString
    }

    client.stop(); //stop client
    Serial.println("client disconnected.");
    Serial.println("Data from server captured in readString:");
    Serial.println();
    Serial.print(readString); //prints readString to serial monitor
}

```



```

Serial.println();
Serial.println("End of readString");
Serial.println("=====");
Serial.println();

if (readString.indexOf(arrayon[8]) > 0)
{
    digitalWrite(38, HIGH);
    readString = ""; //clear readString variable
}
else if (readString.indexOf(arrayoff[8]) > 0)
{
    digitalWrite(38, LOW);
}

}
//=====

//=====
void settings()
{
    s[8] = 0;
    r[5] = 0;
    t[5] = 0;
    LightBottomLimit[5] = 0;
    LightUpperLimit[5] = 0;
    TempBottomLimit[5] = 0;
    TempUpperLimit[5] = 0;
    x = 0;
    y = 0;

    while (whichKey != 'B')
    {
        whichKey = myKeypad.getKey();
        while (whichKey != 'A' && whichKey != 'B')
        {
            whichKey = myKeypad.getKey();
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Choose Setting!");
            lcd.setCursor(0, 2);
            lcd.print("A.Cahaya B.Suhu");
            delay(100);
        }

        //Setting Sensor Cahaya-----
        if (whichKey == 'A')
        {

```

```

        whichKey = myKeypad.getKey();
        while (whichKey != '1' && whichKey != '2' && whichKey != '3' && whichKey !=
'4' && whichKey != '5')
        {
            whichKey = myKeypad.getKey();
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Choose Sensor!");
            lcd.setCursor(0, 2);
            lcd.print("1.C1 2.C2 3.C3");
            lcd.setCursor(0, 3);
            lcd.print("4.C4 5.C5");
            delay(100);
        }
        y = whichKey - '0'; //convert char value to integer
        s[y - 1] = y;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Settings for");
        lcd.setCursor(0, 1);
        lcd.print("Sensor C");
        lcd.setCursor(8, 1);
        lcd.print(y);
        delay(2000);

        whichKey = myKeypad.getKey();
        while (whichKey != '1' && whichKey != '2' && whichKey != '3' && whichKey !=
'4' && whichKey != '5' && whichKey != '6' && whichKey != '7' && whichKey != '8')
        {
            whichKey = myKeypad.getKey();
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Link Channel");
            lcd.setCursor(0, 1);
            lcd.print("1 2 3 4 5 6 7 8");
            delay(100);
        }
        x = whichKey - '0'; //convert char value to integer
        r[y - 1] = x; //Hubungkan sensor dengan channel relay
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("C linked to");
        lcd.setCursor(1, 0);
        lcd.print(y);
        lcd.setCursor(0, 1);
        lcd.print("Channel ");
        lcd.print(r[y - 1]);
        delay(1000);
        whichKey = myKeypad.getKey();

```

```

while (whichKey != '#')
{
    int SensorValue = analogRead(y - 1);
    SensorValue = ((SensorValue - 15) / 446.8) * 100;
    if (SensorValue <= 0)
        SensorValue = 0;
    if (SensorValue >= 100)
        SensorValue = 100;
    whichKey = myKeypad.getKey();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Bottom Limit C");
    lcd.print(y);
    lcd.print(":");
    lcd.setCursor(0, 1);
    lcd.print("SELECT ");
    lcd.print(SensorValue);
    lcd.print("%");
    delay(100);
}
int      SensorValue = analogRead(y - 1);
SensorValue = ((SensorValue - 15) / 446.8) * 100;
if (SensorValue <= 0)
    SensorValue = 0;
if (SensorValue >= 100)
    SensorValue = 100;
LightBottomLimit[y - 1] = SensorValue; //Mengambil nilai bottom limit sensor C1
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Bottom Limit C");
lcd.print(y);
lcd.setCursor(0, 1);
lcd.print("is set to  ");
lcd.print(LightBottomLimit[y - 1]);
lcd.print("%");
delay(2000);
whichKey = myKeypad.getKey();
while (whichKey != '#')
{
    int SensorValue = analogRead(y - 1);
    SensorValue = ((SensorValue - 15) / 446.8) * 100;
    if (SensorValue <= 0)
        SensorValue = 0;
    if (SensorValue >= 100)
        SensorValue = 100;
    whichKey = myKeypad.getKey();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Upper Limit C");

```

```

    lcd.print(y);
    lcd.print(":");
    lcd.setCursor(0, 1);
    lcd.print("SELECT ");
    lcd.print(SensorValue);
    lcd.print("%");
    delay(100);
}
SensorValue = analogRead(y - 1);
SensorValue = ((SensorValue - 15) / 446.8) * 100;
if (SensorValue <= 0)
    SensorValue = 0;
if (SensorValue >= 100)
    SensorValue = 100;
LightUpperLimit[y - 1] = SensorValue; //Mengambil nilai upper limit sensor C1
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Upper Limit C");
lcd.print(y);
lcd.setCursor(0, 1);
lcd.print("is set to ");
lcd.print(LightUpperLimit[y - 1]);
lcd.print("%");
delay(2000);
}
//Setting Sensor Cahaya-----

//Setting Sensor Suhu-----
if (whichKey == 'B')
{
    whichKey = myKeypad.getKey();
    while (whichKey != '1' && whichKey != '2' && whichKey != '3')
    {
        whichKey = myKeypad.getKey();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Choose Sensor!");
        lcd.setCursor(0, 2);
        lcd.print("1.S1 2.S2 3.S3");
        delay(100);
    }
    y = whichKey - '0'; //convert char value to integer
    t[y - 1] = y;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Settings for");
    lcd.setCursor(0, 1);
    lcd.print("Sensor S");
    lcd.setCursor(8, 1);

```

```

lcd.print(y);
delay(2000);

whichKey = myKeypad.getKey();
while (whichKey != '1' && whichKey != '2' && whichKey != '3' && whichKey !=
'4' && whichKey != '5' && whichKey != '6' && whichKey != '7' && whichKey != '8')
{
    whichKey = myKeypad.getKey();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Link Channel");
    lcd.setCursor(0, 1);
    lcd.print("1 2 3 4 5 6 7 8");
    delay(100);
}
x = whichKey - '0'; //convert char value to integer
r[y + 4] = x; //Hubungkan sensor dengan channel relay
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("S linked to");
lcd.setCursor(1, 0);
lcd.print(y);
lcd.setCursor(0, 1);
lcd.print("Channel ");
lcd.print(r[y + 4]);
delay(1000);

whichKey = myKeypad.getKey();
lcd.clear();
strTemp = "";
while (whichKey != '#')
{
    whichKey = myKeypad.getKey();
    lcd.setCursor(0, 0);
    lcd.print("Set Bottom Limit S");
    lcd.print(y);
    lcd.print(".");
    if (whichKey)
    {
        strTemp += whichKey;
        lcd.setCursor(0, 2);
        lcd.print(strTemp);
        lcd.print("C");
    }
    delay(100);
}
TempBottomLimit[y - 1] = strTemp.toInt(); //Mengambil nilai bottom limit sensor
C1
lcd.clear();

```

```

lcd.setCursor(0, 0);
lcd.print("Bottom Limit S");
lcd.print(y);
lcd.setCursor(0, 1);
lcd.print("is set to  ");
lcd.print(TempBottomLimit[y - 1]);
lcd.print("C");
delay(2000);

```

```

whichKey = myKeypad.getKey();
lcd.clear();
strTemp = "";
while (whichKey != '#')
{
    whichKey = myKeypad.getKey();
    lcd.setCursor(0, 0);
    lcd.print("Set Upper Limit S");
    lcd.print(y);
    lcd.print(":");
    if (whichKey)
    {
        strTemp += whichKey;
        lcd.setCursor(0, 2);
        lcd.print(strTemp);
        lcd.print("C");
    }
    delay(100);
}
TempUpperLimit[y - 1] = strTemp.toInt(); //Mengambil nilai bottom limit sensor

```

C1

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Upper Limit S");
lcd.print(y);
lcd.setCursor(0, 1);
lcd.print("is set to  ");
lcd.print(TempUpperLimit[y - 1]);
lcd.print("C");
delay(2000);
}

```

//Setting Sensor Suhu-----

```

whichKey = myKeypad.getKey();
while (whichKey != 'A' && whichKey != 'B')
{
    whichKey = myKeypad.getKey();
    lcd.clear();
    lcd.setCursor(0, 0);
}

```

```

        lcd.print("Set Another Sensor?");
        lcd.setCursor(0, 2);
        lcd.print("A.YES   B.NO");
        delay(100);
    }
    if (whichKey == 'B')
    {
        digitalWrite(RelayChannel[0], LOW);
        digitalWrite(RelayChannel[1], LOW);
        digitalWrite(RelayChannel[2], LOW);
        digitalWrite(RelayChannel[3], LOW);
        digitalWrite(RelayChannel[4], LOW);
        digitalWrite(RelayChannel[5], LOW);
        digitalWrite(RelayChannel[6], LOW);
        digitalWrite(RelayChannel[7], LOW);
        digitalWrite(RelayChannel[8], LOW);
    }
}
}
//=====

//=====
void AutoMode()
{
    LightValue[1] = analogRead(A0);
    LightValue[1] = ((LightValue[1] - 15) / 446.8) * 100;
    if (LightValue[1] <= 0)
        LightValue[1] = 0;
    if (LightValue[1] >= 100)
        LightValue[1] = 100;
    LightValue[2] = analogRead(A1);
    LightValue[2] = ((LightValue[2] - 15) / 446.8) * 100;
    if (LightValue[2] <= 0)
        LightValue[2] = 0;
    if (LightValue[2] >= 100)
        LightValue[2] = 100;
    LightValue[3] = analogRead(A2);
    LightValue[3] = ((LightValue[3] - 15) / 446.8) * 100;
    if (LightValue[3] <= 0)
        LightValue[3] = 0;
    if (LightValue[3] >= 100)
        LightValue[3] = 100;
    LightValue[4] = analogRead(A3);
    LightValue[4] = ((LightValue[4] - 15) / 446.8) * 100;
    if (LightValue[4] <= 0)
        LightValue[4] = 0;
    if (LightValue[4] >= 100)
        LightValue[4] = 100;
}

```

```

LightValue[5] = analogRead(A4);
LightValue[5] = ((LightValue[5] - 15) / 446.8) * 100;
if (LightValue[5] <= 0)
    LightValue[5] = 0;
if (LightValue[5] >= 100)
    LightValue[5] = 100;

sensors1.requestTemperatures(); // Send the command to get temperatures
sensors2.requestTemperatures();
sensors3.requestTemperatures();

TempValue[1] = sensors1.getTempCByIndex(0);
TempValue[2] = sensors2.getTempCByIndex(0);
TempValue[3] = sensors3.getTempCByIndex(0);

//Activity Based on Light Sensor -----

if (LightValue[s[0]] <= LightBottomLimit[0])
    digitalWrite(RelayChannel[r[0]], HIGH);
else if (LightValue[s[0]] >= LightUpperLimit[0])
    digitalWrite(RelayChannel[r[0]], LOW);

if (LightValue[s[1]] <= LightBottomLimit[1])
    digitalWrite(RelayChannel[r[1]], HIGH);
else if (LightValue[s[1]] >= LightUpperLimit[1])
    digitalWrite(RelayChannel[r[1]], LOW);

if (LightValue[s[2]] <= LightBottomLimit[2])
    digitalWrite(RelayChannel[r[2]], HIGH);
else if (LightValue[s[2]] >= LightUpperLimit[2])
    digitalWrite(RelayChannel[r[2]], LOW);

if (LightValue[s[3]] <= LightBottomLimit[3])
    digitalWrite(RelayChannel[r[3]], HIGH);
else if (LightValue[s[3]] >= LightUpperLimit[3])
    digitalWrite(RelayChannel[r[3]], LOW);

if (LightValue[s[4]] <= LightBottomLimit[4])
    digitalWrite(RelayChannel[r[4]], HIGH);
else if (LightValue[s[4]] >= LightUpperLimit[4])
    digitalWrite(RelayChannel[r[4]], LOW);

//-----

//Activity Based On Temperature Sensor -----

if (TempValue[t[0]] <= TempBottomLimit[0])
    digitalWrite(RelayChannel[r[5]], LOW);
else if (TempValue[t[0]] >= TempUpperLimit[0])

```



```

        digitalWrite(RelayChannel[r[5]], HIGH);

    if (TempValue[t[1]] <= TempBottomLimit[1])
        digitalWrite(RelayChannel[r[6]], LOW);
    else if (TempValue[t[1]] >= TempUpperLimit[1])
        digitalWrite(RelayChannel[r[6]], HIGH);

    if (TempValue[t[2]] <= TempBottomLimit[2])
        digitalWrite(RelayChannel[r[7]], LOW);
    else if (TempValue[t[2]] >= TempUpperLimit[2])
        digitalWrite(RelayChannel[r[7]], HIGH);
}
//=====================================================

//=====================================================
void ManualMode()
{
    for (int n = 0; n < 8; n++)
    {
        if (readString.indexOf(arrayon[n]) > 0)
        {
            digitalWrite((2 * n) + 22, HIGH);
        }
    }

    for (int f = 0; f < 8; f++)
    {
        if (readString.indexOf(arrayoff[f]) > 0)
        {
            digitalWrite((2 * f) + 22, LOW);
        }
    }
    readString = ""; //clear readString variable
}
//=====================================================

//=====================================================
void MonitorCahaya()
{
    lcd.clear();
    whichKey = myKeypad.getKey();
    while (whichKey != '*')
    {
        whichKey = myKeypad.getKey();
        lcd.setCursor(0, 0);
        lcd.print("Monitoring Cahaya");

        LightValue[1] = analogRead(A0);
    }
}

```

```

LightValue[1] = ((LightValue[1] - 15) / 446.8) * 100;
if (LightValue[1] <= 0)
    LightValue[1] = 0;
if (LightValue[1] >= 100)
    LightValue[1] = 100;
LightValue[2] = analogRead(A1);
LightValue[2] = ((LightValue[2] - 15) / 446.8) * 100;
if (LightValue[2] <= 0)
    LightValue[2] = 0;
if (LightValue[2] >= 100)
    LightValue[2] = 100;
LightValue[3] = analogRead(A2);
LightValue[3] = ((LightValue[3] - 15) / 446.8) * 100;
if (LightValue[3] <= 0)
    LightValue[3] = 0;
if (LightValue[3] >= 100)
    LightValue[3] = 100;
LightValue[4] = analogRead(A3);
LightValue[4] = ((LightValue[4] - 15) / 446.8) * 100;
if (LightValue[4] <= 0)
    LightValue[4] = 0;
if (LightValue[4] >= 100)
    LightValue[4] = 100;
LightValue[5] = analogRead(A4);
LightValue[5] = ((LightValue[5] - 15) / 446.8) * 100;
if (LightValue[5] <= 0)
    LightValue[5] = 0;
if (LightValue[5] >= 100)
    LightValue[5] = 100;

lcd.setCursor(0, 1);
lcd.print("          ");
lcd.setCursor(0, 1);
lcd.print("C1:");
lcd.print(LightValue[1]);
lcd.print("%");

lcd.setCursor(0, 2);
lcd.print("          ");
lcd.setCursor(0, 2);
lcd.print("C2:");
lcd.print(LightValue[2]);
lcd.print("%");

lcd.setCursor(0, 3);
lcd.print("          ");
lcd.setCursor(0, 3);
lcd.print("C3:");
lcd.print(LightValue[3]);

```

```

        lcd.print("%");

        lcd.setCursor(9, 1);
        lcd.print("    ");
        lcd.setCursor(9, 1);
        lcd.print("C4:");
        lcd.print(LightValue[4]);
        lcd.print("%");

        lcd.setCursor(9, 2);
        lcd.print("    ");
        lcd.setCursor(9, 2);
        lcd.print("C5:");
        lcd.print(LightValue[5]);
        lcd.print("%");
    }
}

//=====================================================

//=====================================================
void MonitorSuhu()
{
    lcd.clear();
    whichKey = myKeypad.getKey();
    while (whichKey != '*')
    {
        whichKey = myKeypad.getKey();
        lcd.setCursor(0, 0);
        lcd.print("Monitoring Suhu");

        sensors1.requestTemperatures(); // Send the command to get temperatures
        sensors2.requestTemperatures();
        sensors3.requestTemperatures();

        TempValue[1] = sensors1.getTempCByIndex(0);
        TempValue[2] = sensors2.getTempCByIndex(0);
        TempValue[3] = sensors3.getTempCByIndex(0);

        lcd.setCursor(0, 1);
        lcd.print("    ");
        lcd.setCursor(0, 1);
        lcd.print("S1:");
        lcd.print(TempValue[1]);
        lcd.print("C");

        lcd.setCursor(0, 2);
        lcd.print("    ");

```

```

    lcd.setCursor(0, 2);
    lcd.print("S2:");
    lcd.print(TempValue[2]);
    lcd.print("C");

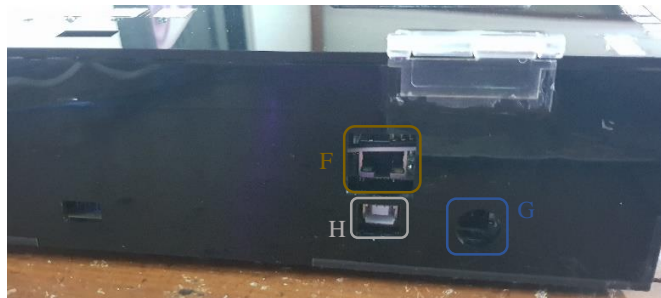
    lcd.setCursor(0, 3);
    lcd.print(" ");
    lcd.setCursor(0, 3);
    lcd.print("S3:");
    lcd.print(TempValue[3]);
    lcd.print("C");
    whichKey = myKeypad.getKey();
}
}
//=====

```

## LAMPIRAN D

### Buku Manual Modul Sistem Otomasi Bangunan

#### I. Komponen-Komponen Pada *Control Unit* Sistem Otomasi Bangunan:



- A. Indikator Power berupa LED merah yang menyala saat sistem telah aktif.
- B. Indikator Keadaan Channel berupa LED hijau yang dilengkapi nomor sebagai identitas tiap *channel*.
- C. Slot Channel sebagai tempat masuknya pucuk obeng untuk melonggarkan dan mengeratkan penghubung antara terminal *channel control unit* dengan kabel keluaran.

- D. Slot Daya Switching sebagai tempat masuknya pucuk obeng untuk melonggarkan dan mengeratkan penghubung antara terminal *switching power control unit* dengan kabel masukan.  
**Note: Switching power adalah sumber listrik yang akan dikontrol oleh control unit.**
- E. Slot Sensor yang menjadi tempat penghubung sensor dengan *control unit*.
- F. Slot Ethernet sebagai tempat dihubungkannya kabel ethernet dengan *control unit*.
- G. Slot Daya Sistem sebagai tempat dihubungkannya kabel *power* dari adaptor 12V dengan *control unit*.
- H. Slot Pemrograman Sistem sebagai tempat dihubungkannya kabel *downloader* Arduino dengan *control unit*.
- I. Keypad sebagai media untuk *user* melakukan pengaturan sensor dan memonitor keadaan sensor.
- J. Display sebagai media pemberi informasi-informasi yang dibutuhkan kepada *user*. Informasi-informasi tersebut terdiri dari:
- Tulisan “PING” sebagai indikator sistem telah memperbarui data dari internet.
  - Mode operasi yang sedang aktif (*Auto Mode* atau *Manual Mode*).
  - Panduan untuk *user* saat melakukan fungsi pengaturan.
  - Nilai-nilai pembacaan sensor suhu dan sensor intensitas cahaya saat dilakukan fungsi *monitoring*.

## II. *Standard Operating Procedure (SOP)*

### A. *General*

1. Hubungkan sumber daya melalui adaptor 12V ke *slot* daya sistem.
2. Hubungkan *router internet* dengan *control unit* dengan kabel ethernet melalui *slot ethernet*.
3. Hubungkan sumber daya yang ingin dikontrol oleh *control unit* ke *slot* daya *switching* dengan melonggarkan sekrup terminal, menghubungkan kabel daya dengan terminal, kemudian mengeratkan kembali sekrup terminal.
4. Hubungkan terminal *channel-channel* keluaran dengan kabel jalur kelistrikan sesuai keinginan, dengan cara melonggarkan sekrup terminal, menghubungkan kabel jalur kelistrikan yang diinginkan, dan mengeratkan kembali sekrup terminal.
5. Untuk mengubah mode dari manual ke auto atau sebaliknya, tekan tombol “mode” (terlihat pada gambar A) pada aplikasi android.
6. Untuk melakukan kontrol manual, tekan tombol kontrol manual pada aplikasi android (terlihat pada gambar A) sesuai *channel relay* yang ingin dikontrol.
7. Untuk memasuki fungsi pengaturan, tekan tombol “**STAR**” pada *keypad control unit*.
8. Untuk melakukan *monitoring* sensor cahaya, tekan tombol “**A**” pada *keypad control unit*. Kemudian tekan tombol “**STOP**” untuk mengakhiri *monitoring*.
9. Untuk melakukan *monitoring* sensor suhu, tekan tombol “**B**” pada *keypad control unit*. Kemudian tekan tombol “**STOP**” untuk mengakhiri *monitoring*.

### B. *Fungsi Pengaturan*

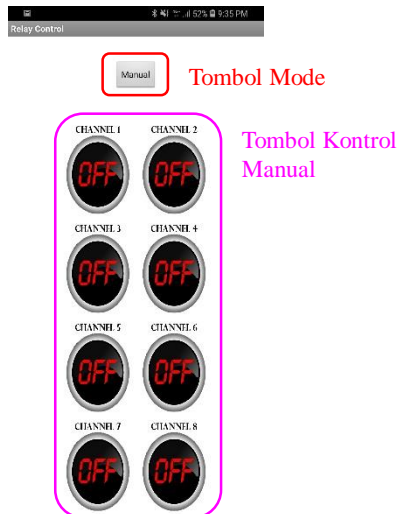
Setelah masuk ke dalam fungsi pengaturan, *user* akan memilih jenis sensor yang akan diatur (sensor cahaya atau suhu) dengan menekan tombol “A” atau “B” pada *keypad* sesuai perintah pada *display*.

- Pengaturan sensor cahaya

1. Tekan salah satu dari tombol “1”, “2”, “3”, “4”, atau “5” untuk memilih sensor cahaya yang ingin diatur seperti yang ditampilkan pada *display*.
  2. Pada menu setelahnya, tekan salah satu dari tombol “1”, “2”, “3”, “4”, “5”, “6”, “7”, atau “8” untuk memilih *channel* keluaran yang akan dihubungkan dengan sensor yang sedang diatur.
  3. Menu selanjutnya berfungsi untuk menentukan nilai batas bawah dari sensor cahaya yang sedang diatur. *Display* akan menampilkan nilai sensor cahaya yang sedang diatur secara *real-time*. Tekan tombol “STAR” pada *keypad* untuk mengambil nilai yang sedang ditampilkan sebagai nilai batas bawah sensor.
  4. Menu setelahnya berfungsi untuk menentukan nilai batas atas dari sensor cahaya yang sedang diatur. *Display* akan menampilkan nilai sensor cahaya yang sedang diatur secara *real-time*. Tekan tombol “STAR” pada *keypad* untuk mengambil nilai yang sedang ditampilkan sebagai nilai batas atas sensor.
  5. Pengaturan sensor cahaya selesai.
- Pengaturan sensor cahaya
    1. Tekan salah satu dari tombol “1”, “2”, atau “3” untuk memilih sensor suhu yang ingin diatur seperti yang ditampilkan pada *display*.
    2. Pada menu setelahnya, tekan salah satu dari tombol “1”, “2”, “3”, “4”, “5”, “6”, “7”, atau “8” untuk memilih *channel* keluaran yang akan dihubungkan dengan sensor yang sedang diatur.
    3. Menu selanjutnya berfungsi untuk menentukan nilai batas bawah dari sensor suhu yang sedang diatur. Tekan tombol angka pada *keypad* untuk menulis nilai sensor suhu yang akan dijadikan nilai batas bawah sensor suhu dalam satuan celcius. Kemudian tekan tombol “STAR” untuk menyimpan angka tersebut.



4. Menu setelahnya berfungsi untuk menentukan nilai batas atas dari sensor suhu yang sedang diatur. Tekan tombol angka pada *keypad* untuk menulis nilai sensor suhu yang akan dijadikan nilai batas atas sensor suhu dalam satuan celcius. Kemudian tekan tombol “**STAR**” untuk menyimpan angka tersebut.
5. Pengaturan sensor suhu selesai.



**Gambar A.** Tampilan Aplikasi Android



**Gambar B.** Tampilan Keypad dan Display

.....*Halaman ini sengaja dikosongkan*.....

## BIODATA PENULIS



Penulis dilahirkan di Jakarta pada tanggal 24 Maret 1996, yang adalah anak kedua dari tiga bersaudara. Penulis menyelesaikan kegiatan pendidikan dasar di SD Strada Wiyatasana Pasar Minggu, kemudian melanjutkan pendidikan di SMP Strada Marga Mulia Pasar Minggu, dan setelahnya di SMA Negeri 6 Jakarta. Pada tahun 2014 penulis diterima sebagai mahasiswa di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Dalam

masa perkuliahan, penulis aktif dalam berbagai macam kepanitian, kegiatan sosial, dan organisasi (khususnya badan eksekutif). Penulis juga turut berpartisipasi sebagai asisten laboratorium dan praktikum di bidang studi elektronika.

Email : rephamstrabena@gmail.com  
HP/WhatsApp : 081380121996  
ID Line : gagax  
Facebook : Rephando Amstrabena